REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 21-12-2015	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 18-Aug-2010 - 17-Jan-2015

4. TITLE AND SUBTITLE
Final Report: Inference for Identity Management

5a. CONTRACT NUMBER
W911NF-10-1-0387

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Carlo Tomasi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Duke University
2200 West Main Street
Suite 710
Durham, NC 27705 -4010

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
58350-MA.29

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

We developed a mathematical formulation and a set of algorithms that make significant strides towards a practical system for identity management. At the core of the formulation lies a single binary integer program that describes the key data association problem: Nodes in a graph correspond to observations, and edges are weighted with correlation measures that quantify positive or negative evidence for the hypothesis that two nodes correspond to observations of the same person. The binary integer program defines a partition of the nodes into sets that are meant to correspond to distinct identities. Solving this problem is NP-hard, and we developed a problem decomposition

15. SUBJECT TERMS
Computer vision, correlation clustering, identity management

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Carlo Tomasi
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 919-660-6539

Report Title

Final Report: Inference for Identity Management

ABSTRACT

We developed a mathematical formulation and a set of algorithms that make significant strides towards a practical system for identity management. At the core of the formulation lies a single binary integer program that describes the key data association problem: Nodes in a graph correspond to observations, and edges are weighted with correlation measures that quantify positive or negative evidence for the hypothesis that two nodes correspond to observations of the same person. The binary integer program defines a partition of the nodes into sets that are meant to correspond to distinct identities. Solving this problem is NP-hard, and we developed a problem decomposition method that, while losing optimality guarantees, show good empirical performance at near frame-rate. To evaluate our method and establish a baseline for future work by us and others, we developed a large video data set with more than 1 million frames and more than 2000 identities observed from eight cameras placed on the campus of Duke University. The data set is fully annotated, and a 3D trajectory is available for each person in every frame from every camera. We also formulated a new methodology for performance evaluation in identity management.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received		Paper
04/26/2011	1.00	Steve Gu, Carlo Tomasi. Branch and Track, , (04 2011): . doi:
04/26/2011	3.00	Steve Gu, Ying Zheng, Carlo Tomasi. Efficient visual object tracking with online nearest neighbor classifier, , (04 2011): . doi:
04/26/2011	2.00	Joaquin Salas, Carlo Tomasi. People Detection Using Color and Depth Images, , (04 2011): . doi:
12/21/2015	27.00	Carlo Tomasi, Joaquin Salas. A linear system form solution to compute the local space average color, Machine Vision and Applications, (03 2013): 1555. doi: 10.1007/s00138-013-0494-0

TOTAL: 4

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
07/17/2014	26.00	Susanna Ricco, Carlo Tomasi. Video Motion for Every Visible Point, 2013 IEEE International Conference on Computer Vision (ICCV). 01-DEC-13, Sydney, Australia. : ,
08/12/2013	23.00	Carlo Tomasi, Susanna Ricco. Simultaneous comopaction and factorization of sparse image motion matrices, European Conference on Computer Vision. 07-OCT-12, . : ,
08/12/2013	25.00	Steve Gu, Ying Zheng, Carlo Tomasi. Nested pictorial structures, European Conference on Computer Vision. 07-OCT-12, . : ,
08/12/2013	24.00	Steve Gu, Carlo Tomasi, Ying Zheng. Fast tiered labeling with topological priors, European Conference on Computer Vision. 07-OCT-12, . : ,
08/16/2012	15.00	Susanna Ricco, Carlo Tomasi. Dense Lagrangian motion estimation with occlusions, IEEE Conference on Computer Vision and Pattern Recognition. 16-JUN-12, . : ,
08/16/2012	19.00	Steve Gu, Carlo Tomasi, Ying Zheng. Topological persistence on a Jordan curve, IEEE International Conference on Acoustics, Speech, and Signal Processing. 27-MAR-12, . : ,
08/16/2012	18.00	Ying Zheng, Carlo Tomasi, Steve Gu. Shape from point features, IEEE International Conference on Acoustics, Speech, and Signal Processing. 27-MAR-12, . : ,
08/16/2012	17.00	Steve Gu, Ying Zheng, Carlo Tomasi. Oscillation regularization, IEEE International Conference on Acoustics, Speech, and Signal Processing. 27-MAR-12, . : ,
08/16/2012	16.00	Steve Gu, Ying Zheng, Carlo Tomasi. Twisted window search for efficient shape localization, IEEE Conference on Computer Vision and Pattern Recognition. 16-JUN-12, . : ,
08/26/2011	12.00	Steve Gu, Ying Zheng, Carlo Tomasi. Detecting motion synchrony by video tubes, International Conference on Multimedia. 28-NOV-11, . : ,
08/26/2011	13.00	Steve Gu, Ying Zheng, Carlo Tomasi. Linear time offline tracking and lower envelope algorithms, International Conference on Computer Vision. 06-NOV-11, . : ,
08/26/2011	14.00	Steve Gu, Ying Zheng, Carlo Tomasi. Extended pairwise potentials, IEEE Cnference on Computer Vision and Pattern Recognition. 21-JUN-11, . : ,
12/21/2015	28.00	Ergys Ristani, Carlo Tomasi. Tracking multiple people online and in real time, Asian Conference on Computer Vision . 01-NOV-14, . : ,

TOTAL: **13**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
04/27/2011	4.00	Ying Zheng, Steve Gu, Carlo Tomasi. Detecting motion synchrony by video tubes, (04 2011)
04/27/2011	5.00	Steve Gu, Ying Zheng, Carlo Tomasi. Free-Shape Localization, (04 2011)
04/27/2011	6.00	Steve Gu, Ying Zheng, Carlo Tomasi. Saliency and localization by nested window search, (04 2011)
04/27/2011	7.00	Ying Zheng, Steve Gu, Carlo Tomasi. Topological persistence on a Jordan curve, (04 2011)
04/27/2011	8.00	Ying Zheng, Steve Gu, Carlo Tomasi. Structural symmetry, (04 2011)
04/27/2011	9.00	Steve Gu, Ying Zheng, Carlo Tomasi. Extended pairwise potentials, (04 2011)
04/27/2011	10.00	Steve Gu, Ying Zheng, Carlo Tomasi. Geometric modes and clusters, (04 2011)
04/27/2011	11.00	Steve Gu, Ying Zheng, Carlo Tomasi. Linear time offline tracking and lower envelope algorithms, (04 2011)
08/16/2012	21.00	Ying Zheng, Steve Gu, Carlo Tomasi. Fast tiered labeling with topological priors, Under Review (11 2011)
08/16/2012	20.00	Susanna Ricco, Carlo Tomasi. Simultaneous compaction and factorization of sparse image motion matrices, Under Review (12 2011)
08/16/2012	22.00	Steve Gu, Ying Zheng, Carlo Tomasi. Nested pictorial structures, Under Review (11 2012)

TOTAL: **11**

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Ergys Ristani	0.50	
FTE Equivalent:	**0.50**	
Total Number:	**1**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

See Attachment

Technology Transfer

Final Progress Report
Sensing and Efficient Inference for Identity Management

Agreement Number W911NF-10-1-0387, Proposal Number 58350-MA
Period from August 18, 2010 to January 17, 2015

Carlo Tomasi, Computer Science Department
Duke University, Durham, NC 27708
tomasi@cs.duke.edu

December 20, 2015

Abstract

Identity management entails monitoring people in a shopping mall, airport, or other large area with a set of video cameras, and determining automatically who is where at all times. We developed a mathematical formulation and a set of algorithms that make significant strides towards a practical system for identity management. At the core of the formulation lies a single binary integer program that describes the key data association problem: Nodes in a graph correspond to observations, and edges are weighted with correlation measures that quantify positive or negative evidence for the hypothesis that two nodes correspond to observations of the same person. The binary integer program defines a partition of the nodes into sets that are meant to correspond to distinct identities. Solving this problem is NP-hard, and we developed a problem decomposition method that, while losing optimality guarantees, show good empirical performance at near frame-rate. To evaluate our method and establish a baseline for future work by us and others, we developed a large video data set with more than 1 million frames and more than 2000 identities observed from eight cameras placed on the campus of Duke University. The data set is fully annotated, and a 3D trajectory is available for each person in every frame from every camera. We also formulated a new methodology for performance evaluation in identity management.

To achieve these results, one principal investigator worked with 9 students to publish 17 articles in top venues in computer vision and image processing. The sum of this work constitutes a significant contribution to the field. In addition, three of the students graduated with a PhD based on their work under the project. Two more earned a Master degree, an additional two are still working towards their PhDs, and two undergraduates received their first exposure to advanced computer science research through their work for this project. New collaborations were established with research organizations in Mexico and Italy.

Foreword

The core question addressed by this project is how to monitor a large area such as a shopping mall, airport, or university campus with a set of video cameras and determine automatically who is where at all times. This problem is sometimes called *identity management*. A system to accomplish this task is of obvious usefulness in security, surveillance, crowd control and monitoring, and related applications.

Identity management is difficult for a variety of reasons: Different people may dress or look alike, and, conversely, the same person may look different under different lighting or from different viewpoints. The cameras may not cover the area of interest seamlessly, and people may disappear from one field of view and reappear in another one only after a long interval of time, or not at all. Even more fundamentally: How to distinguish a person from something else? How to track a person reliably in the face of uneven and complex motion, body articulation, occlusions, changes of lighting and viewpoint?

Conceptually, identity management turned out to translate to interesting mathematical problems at all levels of the system. At the core level of data association—which observation relates to what identity—we formulated a Binary Integer Program that captures simply and precisely the overall structure of the problem. At the level of visual tracking, we proposed a new Lagrangian model of image motion that describes the trajectories of every pixel in every frame of a long video sequence. At the lowest levels of image processing and motion analysis, we devised new lighting models, person detection algorithms that combine information from color and depth cameras, new ways to describe image shape, pictorial structures that accelerate person detection, and much more. We marked our exciting journey towards the development of a state-of-the-art theory of identity management with 17 publications in the top computer vision venues (Table 1 on page 2).

For performance evaluation, we developed the largest fully-annotated data set to date, with more than 1 million frames of high-quality video and more than 2000 identities. Dozens of students and friends spent many hours annotating every person in every frame. We are about to make this data set available to the research community, and this contribution alone is likely to accelerate progress in visual tracking and identity management by allowing to compare competing approaches in a fair and thorough manner. We are also proud of a mathematically simple, new methodology that we have developed to evaluate the performance of identity management systems.

This research was a playing field for the intellectual growth of nine students: Zhiqiang Gu, Ying Zheng, and Susanna Ricco graduated with PhD theses directly tied to the project, and are now successfully employed at Google Research and Apple. Ergys Ristani and Cassandra Carley are working towards their PhDs, and have published several articles while funded by this grant. Alan Davidson and Branka Lakic have earned Master degrees in Computer Science with work that advanced various aspects of this project. And undergraduates Trevor Terris and Sterling Dorminey received their first exposure to advanced computer science research thanks to their work with us. Graduate students helped mentor undergraduates, thereby learning a skill that will serve them throughout their professional lives.

To address some of the challenges of our project we reached out to groups in other countries, and particularly with CICATA, the *Centro the Investigación en Ciencia Aplicada y Technología Avanzada* in Querétaro, Mexico, and the *ImageLab* at the University of Modena and Reggio Emilia in Italy. These collaborations continue, and open our minds and those of our students to different ways of thinking and working. They would not have started without ARO funding for this project.

We are therefore very grateful to the Army Research Office for their support of our work, and for the insight and foresight of its officers who saw in a somewhat speculative proposal more than five years ago the potential for transformative research in a field that has in the meantime blossomed—perhaps even in small part thanks to our work—into a thriving subfield of computer vision across the world.

Contents

List of Figures

List of Tables

1 Problem Statement

Several human behavior analysis, monitoring, and surveillance scenarios would benefit from automatic methods that track multiple people through a network of cameras. The *Identity Management* (IM) problem—to determine who is where at all times—is commonly cast as a data association problem: Partition all observations of individuals found in every frame from every camera into a number of sets, one per identity.

Individual observations are typically the outputs from a person detector, and often come in the form of bounding boxes. Detectors are very good nowadays, but they are not perfect, and may miss people or find people where there aren't any. In addition, multiple bounding boxes often cover the same person in any given frame. So the initial observations come with both false positives and false negatives.

To curb computational complexity and handle unbounded time horizons, the IM problem is typically solved over a sliding temporal window and in layers. Specifically—and consistently with the terminology used in most of the literature—a *detection* is a response from a person detector from one camera; a *tracklet* aggregates detections in consecutive frames into short sequences, and a *trajectory* is a sequence of tracklets. Each detection, tracklet, or trajectory is an *observation*. It is constructed from data from a single camera and comes with timing and appearance information which has different formats for different observation types. Tracklets are strung into *identities*, that is, sequences of trajectories from one or more cameras. Each identity is meant to correspond to a single person, and different people to different identities. Tracklets, trajectories, and identities are *aggregates*.

The problem can then be defined as follows: **Observations—at whatever level—are nodes in a graph in which each edge has a *correlation*, that is, a measure of the positive or negative evidence that two observations pertain to the same person. Identity management is the problem of partitioning the nodes of the graph into sets, one set per identity, so that correlations within each set are maximized.** This formulation will be made more precise in the next Section.

What changes between layers is what evidence is encoded in the correlations: For short-term matching, a person's appearance may be captured with image-domain descriptors such as histograms of colors or oriented gradients. In contrast, capturing a person's appearance along a trajectory presents both the challenge of formulating a view-invariant signature and the opportunity for a more nuanced descriptor based on multiple views of the same person. We developed a new, rich way to capture long-term appearance. Also, time and motion may play a strong role in low-level associations, because people move more or less predictably in the short term. For longer-term associations, on the other hand, time and motion are only loosely relevant, since a person may occasionally change direction or speed, or stop while out of sight.

As IM methods solve larger and larger problems, it becomes increasingly important to define performance measures that can handle any and all levels of aggregation consistently. While several such measures have appeared in the literature, they tend to be reliable only either across cameras or within cameras, but rarely for the system as a whole. In particular, current measures fail to satisfactorily address the *truth-to-result matching problem*: A given ground truth trajectory may be claimed by different computed identities that span different, and sometimes even overlapping, time intervals. Which computed identity gets how much credit for that ground-truth trajectory? We developed a simple and general answer to this question, and derived precision and recall measures that match the IM scenario better than existing measures.

We evaluated our system on a standard benchmark, and ran separate experiments to showcase the specific advantages of our new similarity measure. In addition, we experimented with a new data set we built that has more than 1 million frames, fully annotated trajectories, and more than 2000 identities. It consists of 8×85 minutes of 1080p video recorded at 60 frames per second from 8 cameras deployed on the Duke University campus during periods between lectures, when pedestrian traffic is heavy.

- E. Ristani and C. Tomasi. Tracking multiple people online and in real time. *Asian Conference on Computer Vision*, pages 444–459, November 2014.
- S. Ricco and C. Tomasi. Video motion for every visible point. *International Conference on Computer Vision*, pages 456–469, December 2013.
- J. Salas and C. Tomasi. A linear system form solution to compute the local space average color. *Machine Vision and Applications*, pages 1555–1560, October 2013.
- S. Ricco and C. Tomasi. Simultaneous compaction and factorization of sparse image motion matrices. *European Conference on Computer Vision*, pages 456–469, October 2012.
- Y. Zheng, S. Gu, and C. Tomasi. Fast tiered labeling with topological priors. *European Conference on Computer Vision*, pages 587–601, October 2012.
- S. Gu, Y. Zheng, and C. Tomasi. Nested pictorial structures. *European Conference on Computer Vision*, pages 816–827, October 2012.
- S. Ricco and C. Tomasi. Dense Lagrangian Motion Estimation with Occlusions. *IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.
- S. Gu, Y. Zheng, and C. Tomasi. Twisted Window Search for Efficient Shape Localization. *IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.
- S. Gu, Y. Zheng and C. Tomasi. Oscillation regularization. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2012.
- S. Gu, Y. Zheng and C. Tomasi. Shape from point features. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2012.
- Y. Zheng, S. Gu, and C. Tomasi. Topological persistence on a Jordan curve. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 2012.
- Y. Zheng, S. Gu, and C. Tomasi. Detecting motion synchrony by video tubes. *International Conference on Multimedia*, pages 1197-1200, December 2011.
- S. Gu, Y. Zheng, and C. Tomasi. Linear time offline tracking and lower envelope algorithms. *International Conference on Computer Vision*, pages 1840–1846, November 2011.
- S. Gu and C. Tomasi. Branch and Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 271–282, June 2011.
- S. Gu, Y. Zheng, and C. Tomasi. Extended pairwise potentials. *CVPR Workshop on Inference in Graphical Models with Structured Potentials*, June 2011.
- J. Salas and C. Tomasi. People Detection Using Color and Depth Images. *Mexican Conference on Pattern Recognition*, pages 127–135, June 2011.
- S. Gu, Y. Zheng and C. Tomasi. Efficient Visual Object Tracking with Online Nearest Neighbor Classifier. *Asian Conference on Computer Vision*, pages 267–277, December 2010.

Table 1: Publications from this project in reverse chronological order.

2 Summary of Results

We made the following contributions to the core problem of Identity Management (IM):

- A unified formulation that applies both within and between cameras, handling both overlapping and disjoint views seamlessly.
- A new 3D descriptor of trajectory appearance that captures informative appearance details with minimal blurring.
- New, consistent, and simple measures of performance.
- A large, fully-annotated data set.
- Experiments and comparisons.

We describe these contributions in a recent article [65] and in a paper in preparation, and we summarize them in this report.

The contributions above address the core problems of IM. To make a practical system possible, we also had to revisit the standard computer vision components on which such a system must rely: Methods for detecting people in individual frames and tracking them—or something else, for that matter—between frames, as well as basic mathematical models, data structures, and algorithms that make these methods robust and efficient.

These ancillary contributions are discussed in additional papers listed in Table 1. The mot important one, a Lagrangian formulation of image motion, is summarized in the Appendix. We do not use this formulation directly in our work on identity management described in the remainder of this report, mainly because of its high computational cost. Nonetheless, the development of this method has led to many improvements in the motion analysis aspects of our main line of work. In addition, the Lagrangian formulation can lead to person detections that (i) are optimized over an entire video sequence rather than frame by frame, and (ii) delineate the image region occupied by a person, rather than a rectangular bounding box around it. Our identity management experiments show that the contamination of detection descriptors caused by background pixels present in the bounding boxes is a significant source of identification errors. Because of this, we believe that further study is warranted to make the Lagrangian formulation computationally more efficient and therefore practically useful, and we stand by the conceptual and theoretical validity of this formulation.

The following sections cover prior art in Identity Management (IM), our approach, our new performance measure, and experimental results.

2.1 Related Work

Current research on multi-camera identity management relies on improved pedestrian detection [14] and single camera tracking, and most multi-camera IM methods even assume availability of single-camera trajectories at test time [21, 27, 31, 32, 33, 37, 38, 43, 48, 54, 58, 80]. Exceptions [29, 81] solve single and multi-camera in two steps through the same optimization framework.

Camera Placement Information. *Spatial relations between cameras* are explicitly mapped in 3D [31, 80], learnt by tracking known identities [48, 49, 28] or by comparing entry/exit rates across pairs of camera [27, 54, 58], or discovered on-line [43, 81]. Pre-processing methods may fuse data from partially overlapping views [81], while work from completely overlapping and unobstructed views has been regarded as a separate task in the literature [4, 16, 21, 50, 44].

Entry and exit points (EEPs) are sources and sinks of individuals, and may be explicitly estimated as mixtures of Gaussians on the ground plane [31, 27, 54, 58]. Image-space approaches cluster pixels on image boundaries [49] or split the image adaptively to localize EEPs [43]. Online formulations [31] can handle time-varying EEPs.

Travel time information is modeled with Gaussians [49, 80] or in non-parametric ways [31, 43, 48, 54, 58].

Appearance Descriptors. *Color* is summarized with RGB/HSV/YCbCr histograms [27, 31, 32, 33, 38, 43, 48, 49, 54, 80, 81] on foreground masks [33, 37, 38, 43, 49]. *Texture* is modeled through covariance matrices [54], local binary patterns [32] or (P)HOG features [27, 37, 54, 80, 81]. To handle lighting differences between cameras, methods either employ color normalization [27], exemplar based approaches [32] or learn brightness transfer functions [38, 48] even without labeled data [31, 43, 80, 81].

Discriminative power is improved by incorporating *saliency* information [59, 82] or by attaching color and texture features to different *body parts* inside the bounding box [27, 32, 33, 37, 38, 49, 54], either in the image plane task [10, 11, 35] or back-projected onto an articulated [7, 34] or non-articulated 3D body models [8].

Multi-view descriptors are sometimes obtained by averaging descriptors over several frames [32] or by generating random transformations for improved comparisons between descriptors [37]. *Learning* is sometimes employed to weigh features differently for distinct pairs of cameras [32, 54] or to discover target-specific features [27].

Multi-Camera IM Formulations. Spatial, temporal, and appearance information is summarized into *weights* w_{ij} that somehow express the affinity of observation i with observation j. One must eventually decide whether these observations pertain to the same identity or not.

For trajectories, bipartite matching can make this determination for each pair of cameras, but consistency of results across camera pairs is not guaranteed [38]. It can be enforced through global methods in which trajectories are nodes in a graph and the w_{ij}s are edge weights. Path methods compute paths in the graph by maximizing the sum of weights between *consecutive* path nodes, while clique methods maximize the sum of weights on *all* edges in a clique. Path and clique methods apply to tracklets within cameras, or to trajectories across cameras. Some notable contributions are as follows:

	Single-Camera	Cross-Camera	Both
Bipartite	[22, 69, 77]	[27, 32, 37, 54]	—
Path	[16, 47, 63, 79]	[48, 49]	[29, 81]
Clique	[26, 30, 36, 39, 53, 66, 68, 74, 76, 1]	[38]	Ours

In our work, we formulate within and across-camera tracking in a single unified framework, similarly to previous IM flow methods [29, 81]. In contrast with these, our formulation is a clique method and can also handle identities that reappear in the same field of view. Similarly to [38], we consider evidence across the whole network.

Moreover, we describe trajectory appearance with a new signature that, differently from all related work, does not average descriptors appearing at the same location but rather tracks and collects patches inside the detection bounding box over time. We map signatures to a 3D body model only when two signatures are *compared* to each other, and without averaging. This prevents appearance details to be blurred away, exploits information from the entire trajectory, and accounts for viewpoint and pose variation in a flexible and nuanced way.

2.2 Mathematical Problem Formulation

As mentioned earlier, a *detection* is a response from a person detector from one camera; a *tracklet* aggregates detections in consecutive frames into short sequences, and a *trajectory* is a sequence of tracklets. Each detection, tracklet, or trajectory is an *observation*. Tracklets are strung into *identities*, that is, sequences of trajectories from one or more cameras. Each identity is meant to correspond to a single person, and different people to different identities. Tracklets, trajectories, and identities are *aggregates*.

In the following, we first describe our mathematical aggregation framework, which is common to all types of observations, and relies on a graph in which observations are nodes and measures of correlation between observations are edges. Detections, tracklets, and trajectories are different types of observations because of the differences in the amounts of time and variety of viewpoint that they entail. Because of this, node descriptors and edge weights are different for different types of observations, and are described in subsequent sections.

In a typical security or surveillance scenario, observations constitute a flow of unbounded duration. Section 2.2.6 show how we stitch multiple instances of binary integer programs into a processing cascade that can handle an unbounded flow of observations.

2.2.1 Aggregation

In contrast with the literature, we formulate each of the three aggregation steps above as a Binary Integer Program (BIP) of exactly the same format.

Specifically, given edges $(i, j) \in E$ with *signed* weights w_{ij} between observations $i, j \in V$, we set a binary variable x_{ij} to 1 if the observations are to be in the same identity and to 0 otherwise. Weights can be defined between any two observations of the same type, consecutive or not. The graph $G = (V, E)$ need not be complete. For each aggregation problem we then find the

$$\arg\max_{X} \sum_{(i,j)\in E} w_{ij} x_{ij} \tag{1}$$

subject to the following constraint on all triangles of G

$$x_{ij} + x_{jk} \leq 1 + x_{ik} \tag{2}$$

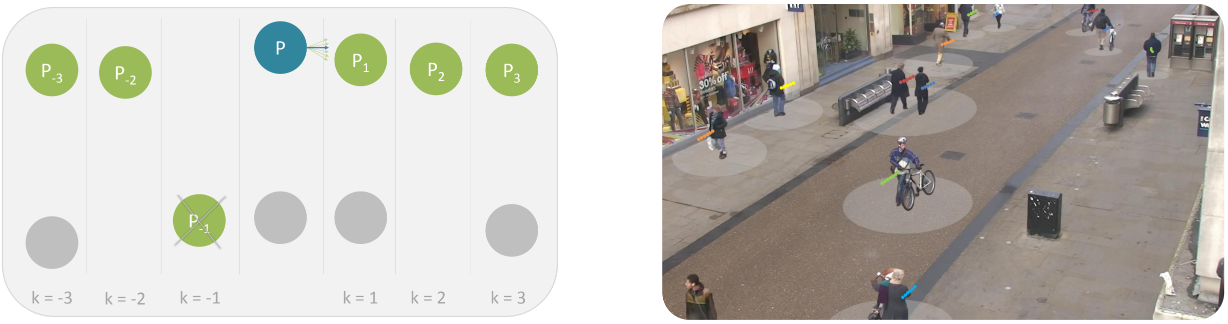
to enforce transitivity: If i and j are in the same identity and so are j and k, then i and k must be in the same identity, too.

Remarkably, the two expressions above capture aggregation precisely, and at all levels of the processing pipeline. Finding an optimal solution to this *correlation clustering* problem is NP-hard [9] and the problem is also hard to approximate [73]. The best known approximation algorithm achieves an approximation ratio of 0.7664 [72], but its semi-definite program formulation makes it slow for practical consideration. These results suggest that one needs to look at the special properties of the multi-person tracking problem to find an efficient solution, as we do in Section 2.2.6.

2.2.2 Detection Descriptors and Correlations

Each person detection $D = (\boldsymbol{\varphi}, \mathbf{p}, t, \mathbf{v})$ is described by its appearance feature $\boldsymbol{\varphi}$, position $\mathbf{p}$, time stamp t, and estimated velocity[1] $\mathbf{v}$. We use an HSV color histogram to describe a person's appearance, but different descriptors can be used with no other modification of the proposed methods.

[1] Velocity is a vector, and its norm is called the *speed*.



(a) (b)

Figure 1: (a) Velocity estimation of the blue detection for $m = 3$. Circles are detections, the horizontal dimension is time, and the vertical one stands for 2D space. Green detections are the nearest detections in space to the blue detection for each k. Detections in grey are not considered for velocity estimation. Detection $\mathbf{p}_{-1}$ is discarded because the speed required to reach the blue detection from it exceeds a predefined limit. The green vectors are the velocities computed for each blue-green detection pair and the blue vector is the estimated velocity. (b) Circles enclose disjoint space-time groups, found from assumed bounds on walking speed.

Co-identity evidence from space and time information comes mainly from the assumption that people are limited in their speed, and reasoning about person speed requires converting image coordinates to world coordinates. To this end, we assume that people move on a planar region and that a homography is available between the world and the image.

The velocity of a detection at position $\mathbf{p}$ in video frame i is estimated as follows. For each frame k in $[i-m, i+m]$ (where m is a small integer) and $k \neq i$, determine the detection $\mathbf{p}_k$ that is nearest (in space) to $\mathbf{p}$. Compute the velocities from each pair $(\mathbf{p}, \mathbf{p}_k)$, and discard those that violate a predefined speed limit. The velocity estimate for the detection at $\mathbf{p}$ is then the component-wise median of the remaining velocities. See Figure 1(a).

Given two detections $D_1 = (\boldsymbol{\varphi}_1, \mathbf{p}_1, t_1, \mathbf{v}_1)$ and $D_2 = (\boldsymbol{\varphi}_2, \mathbf{p}_2, t_2, \mathbf{v}_2)$, we first define two simple space-time and appearance affinity measures for them in $[0, 1]$, and then combine the affinities into a single correlation measure.

Specifically, the space-time affinity of D_1 and D_2 is:

$$s_{st} = \max[1 - \beta\,(e(D_1, D_2) + e(D_2, D_1)), 0] \tag{3}$$

where $e(D_1, D_2) = \|\mathbf{q}_1 - \mathbf{p}_2\|_2$ measures the error between the position $\mathbf{p}_2$ of detection D_2 and the estimated position $\mathbf{q}_1 = \mathbf{p}_1 + \mathbf{v}_1(t_2 - t_1)$ of detection D_1 at time t_2. The parameter β controls how much error we are willing to tolerate. Setting a lower value for β is helpful for handling long occlusions. We use $\beta = 1$.

The appearance affinity between D_1 and D_2 is:

$$s_a = \max[1 - \alpha\, d(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2), 0] \tag{4}$$

where $d(\cdot)$ is a distance function in appearance space. We use the earth mover's distance [67] in our experiments to compare HSV histograms, and set $\alpha = 1$.

A sigmoid function maps affinities to correlations smoothly, except in extreme cases:

$$w = \begin{cases} -\infty & \text{if } s_a s_{st} = 0 \\ +\infty & \text{if } s_a s_{st} = 1 \\ -1 + \frac{2}{1+\exp(-\lambda(s_a s_{st} - \mu))} & \text{otherwise} \end{cases} . \tag{5}$$

The parameter λ determines the width of the transition band between negative and positive correlation, and μ is the value that separates them. We use $\mu = 0.25$, assuming that $s_a = s_t = 0.5$ indicates indifference.

2.2.3 Tracklet Descriptors and Correlations

Tracklets (short trajectories of detections) have somewhat more complex descriptors than individual detections, because they extend over time. Specifically, a tracklet descriptor $\tilde{T} = \{\tilde{\boldsymbol{\varphi}}, \tilde{\mathbf{p}}^s, \tilde{\mathbf{p}}^e, \tilde{t}^s, \tilde{t}^e, \tilde{\mathbf{v}}\}$ contains an appearance feature $\tilde{\boldsymbol{\varphi}}$ that is equal to the median appearance of its detections. The descriptor also contains the start point $\tilde{\mathbf{p}}^s$ and end point $\tilde{\mathbf{p}}^e$ of the tracklet, its start time $\tilde{t}^s$ and end time $\tilde{t}^e$, and its velocity $\tilde{\mathbf{v}}$. Since tracklets are short, we assume that their detections are on a straight line and we approximate the velocity of the tracklets as follows:

$$\tilde{\mathbf{v}} = \frac{\tilde{\mathbf{p}}^e - \tilde{\mathbf{p}}^s}{\tilde{t}^e - \tilde{t}^s} . \tag{6}$$

The definition of appearance affinity remains the same for tracklets, once appearance descriptors are modified as explained earlier. For space-time affinities, the position error $e(\cdot, \cdot)$ is redefined to measure the discrepancy between a tracklet's start point and the estimated start point as determined from the end point of the other tracklet: $e(\tilde{T}_1, \tilde{T}_2) = \|\tilde{\mathbf{q}}_1^s - \tilde{\mathbf{p}}_2^s\|_2$ where $\tilde{\mathbf{q}}_1^s = \tilde{\mathbf{p}}_1^e + \tilde{\mathbf{v}}_1(\tilde{t}_2^s - \tilde{t}_1^e)$.

2.2.4 Trajectory Descriptors and Correlations

Large variations in viewpoint and pose make comparing trajectories harder than comparing individual detections or even tracklets, for which histograms and HOG-like descriptors may be adequate. This problem has been addressed recently [8] by making the comparison in 3D in the context of the so-called re-identification problem. Specifically, color histograms and HOG-like descriptors are back-projected from the images onto a simplified body model called a *sarcophagus* by using camera calibration information, and descriptors from different snapshots are collected and eventually averaged together. The robustness of this method is thus limited to one's ability to project information on the correct vertex of the mesh. In practice, pose changes, bounding box placement errors, and camera calibration inaccuracies often cause the averaging of descriptors that represent different body parts, with consequent loss of detail in the feature representation. These details are crucial, since, say, a bag or the color of one's shoes may distinguish otherwise similarly dressed individuals.

To overcome this limitation, we do not average descriptors on the 3D model. Instead, to remove frame-wise noise, we first average image descriptor histograms over the short *tracks* computed with a Lucas-Kanade tracker, initialized over a grid of points on the foreground mask of the person[78]. When a track is lost, a new point is initialized at the same grid location.

These track averages are more accurate than 3D averages because they do not involve any back-projection, and the tracker vouches for correct correspondence. For each track we store means and variances of descriptor histograms, as well as back-projected body position. A single trajectory T then comes with a set of short tracks $\{\mathbf{p}\}$, each described by a mean descriptor histogram $\boldsymbol{\mu}$, a vector $\boldsymbol{\sigma}$ of bin variances, the number n of frames in the track, and the set $Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ of estimated 3D body positions.

To compare two trajectories, we first define a measure of similarity $s(\mathbf{p}^a, \mathbf{p}^b)$ between two *track* histograms that accounts for the statistical significance of each bin. We then extend this measure to trajectories and convert (non-negative) similarities to (signed) correlations, so we can apply the BIP approach described earlier.

Track Similarity. The means $\boldsymbol{\mu}$ and variances $\boldsymbol{\sigma}$ can be viewed as Gaussian approximations of a distribution over track descriptor histograms $\mathbf{h}$ with independent bins. To compare two histograms $\mathbf{h}^a$ and $\mathbf{h}^b$ we ask: (i) What evidence do corresponding bins $\mathbf{h}_k^a$ and $\mathbf{h}_k^b$ provide for the hypothesis H_0 that the two histograms pertain to the same person? (ii) How significant is this evidence?

We answer the first question by estimating the conditional probability $P(H_0 \mid \mathbf{h}_k^a, \mathbf{h}_k^b)$ through Welch's test [75] of the hypothesis that the two populations with empirical parameters (μ_k^a, σ_k^a) and (μ_k^b, σ_k^b) have equal true means. This test extends to unequal variances the more popular Student's test. The t-values for bin k are computed as

$$t_k(\mathbf{h}_k^a, \mathbf{h}_k^b) = \frac{\mu_k^a - \mu_k^b}{\sqrt{\frac{(\sigma_k^a)^2}{n_k^a} + \frac{(\sigma_k^b)^2}{n_k^b}}}, \tag{7}$$

and then

$$P(\mathbf{h}_k^a, \mathbf{h}_k^b \mid H_0) = 2 \int_{-\infty}^{-|t(\mathbf{h}_k^a, \mathbf{h}_k^b)|} f(t; \upsilon(\mathbf{h}_k^a, \mathbf{h}_k^b)) dt, \tag{8}$$

where $f(t;\upsilon)$ is the probability density function of the t-distribution and the degrees of freedom υ can be estimated in closed form as:

$$\upsilon(\mathbf{h}_k^a, \mathbf{h}_k^b) \approx \frac{\left(\frac{(\sigma_k^a)^2}{n^a} + \frac{(\sigma_k^b)^2}{n^b}\right)^2}{\frac{(\sigma_k^a)^4}{(n^a)^2(n^a-1)} + \frac{(\sigma_k^a)^4}{(n^b)^2(n^b-1)}}\ , \tag{9}$$

which we round to an integer. Assuming independence between the two histograms, Bayes's theorem yields

$$P(H_0|\mathbf{h}_k^a, \mathbf{h}_k^b) = \underbrace{P(\mathbf{h}_k^a, \mathbf{h}_k^b|H_0)}_{p\text{-value or likelihood}} \frac{P(H_0)}{P(\mathbf{h}_k^a)P(\mathbf{h}_k^b)}\ . \tag{10}$$

We estimate $P(\mathbf{h}_k^a)$ and $P(\mathbf{h}_k^b)$ by computing histograms on a regular grid of body location and over the entire training set. The constant term $P(H_0)$ is ignored.

To reflect the greater significance of large bin values for histogram comparison (consideration (ii) above), we weigh each probability above with the size of the smaller of μ_k^a and μ_k^b—a factor reminiscent of histogram intersection—to obtain the *track similarity measure*

$$s(\mathbf{p}^a, \mathbf{p}^b) = \sum_k P(H_0|\mathbf{h}_k^a, \mathbf{h}_k^b)\ \min(\mu_k^a, \mu_k^b)\ . \tag{11}$$

Trajectory Correlation. We extend our track similarity measure to trajectories by defining a track neighborhood by geodesic distance on the 3D body model, and comparing tracks only when they are close to each other in this distance. To prevent double-counting, we perform non-maximum suppression within track neighborhoods.

Specifically, define the distance between tracks $\mathbf{p}_a$ and $\mathbf{p}_b$ as the shortest geodesic distance δ—measured on the 3D model—between them,

$$d(\mathbf{p}^a, \mathbf{p}^b) = \min_{\mathbf{q}^a \in Q^a, \mathbf{q}^b \in Q^b} \delta(\mathbf{q}^a, \mathbf{q}^b) \tag{12}$$

and the neighborhood of track $\mathbf{p}^a$ in trajectory T^b as the set

$$\mathcal{U}(\mathbf{p}^a, T^b) = \{\mathbf{p}^b \in T^b \,|\, d(\mathbf{p}^a, \mathbf{p}^b) \leq \rho\} \tag{13}$$

where ρ is a threshold. We average similarities of $\mathbf{p}^a$ over nearby tracks in T^b to obtain a track-to-trajectory similarity

$$s(\mathbf{p}^a, T^b) = \frac{1}{|\mathcal{U}(\mathbf{p}^a, T^b)|} \sum_{\mathbf{p}^b \in \mathcal{U}(\mathbf{p}^a, T^b)} s(\mathbf{p}^a, \mathbf{p}^b) \ . \tag{14}$$

We then perform non-maximum suppression by retaining a track $\mathbf{p}^a$ iff it achieves the maximum similarity $s(\mathbf{p}^a, T^b)$ in $\mathcal{U}(\mathbf{p}^a, T^a)$. Finally, we convert similarities to correlations

$$c(\mathbf{p}^a, T^b) = s(\mathbf{p}^a, T^b) - Z \tag{15}$$

where Z is determined by cross-validation and denotes the indifference threshold between positive and negative evidence [66], and average the track-to-trajectory correlations to obtain a trajectory correlation

$$c(T^a, T^b) = \frac{\sum_{\mathbf{p}^a \in T^a} c(\mathbf{p}^a, T^b) + \sum_{\mathbf{p}^b \in T^b} c(\mathbf{p}^b, T^a)}{|T^a| + |T^b|} \ . \tag{16}$$

Joint rather than separate normalization in this equation prevents trajectories with a low number of tracks from biasing the correlation score.

With this definition, the 3D model is used mainly to achieve view-invariance through back-projection, while all the track appearance information—averaged in the image plane and only as long as a feature tracker vouches for correct correspondence—is used for trajectory comparisons. Averaging is eventually performed on similarities and correlations, rather than directly on appearance information.

2.2.5 Implementation Details

The side of the patches contained in each track is set to $\frac{1}{20}$ of the bounding box height, so that the same patch always captures the same amount of information independently of the scale. At each frame and from each patch we extract HSV histograms. The color space has been adaptively quantized in 100 bins according to the minimum variance criteria [45] with respect to our training data (see Sec. 2.4.2). On the same data, the parameters $Z = 10$, $\rho = 0.3$m and 0.2m for the non-maximum suppression radius has also been cross-validated. For the optimization framework, the last layer sliding window is 2.5min wide with a stride of half its size.

2.2.6 The Single-Camera Cascade

We now describe a cascade that allows solving the graph partitioning problem defined in Section 2.2.1 approximately and efficiently over a *temporal window* several seconds long and for each camera separately. The longer the window, the longer the occlusions through which identities can be retained. Although we lose theoretical guarantees of optimality, we exploit the special structure of multi-person tracking to decompose

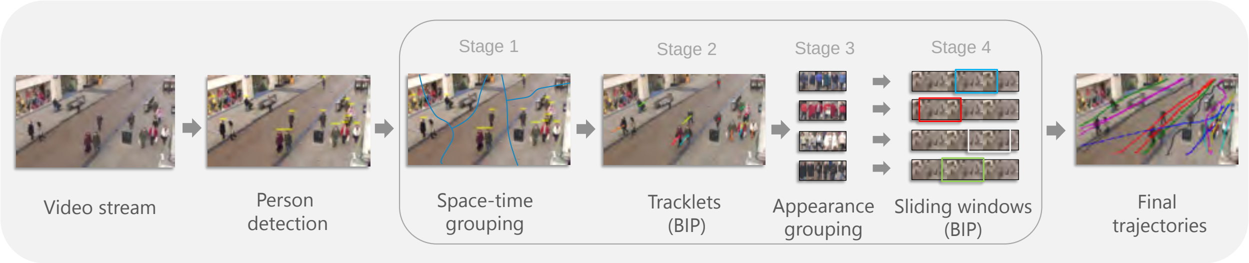


Figure 2: The proposed single-camera processing pipeline from detections to trajectories. One more inter-camera stage aggregates trajectories into identities.

the large BIP problem from Section 2.2.1 into manageable chunks that are unlikely to take us far from the optimal solution. Trajectories output by one such pipeline per camera are then aggregated into identities by one more solution to the basic BIP formulated in section 2.2.1.

Our cascade has two simpler phases divided into two stages each. The first phase partitions detections over short time horizons and results into *tracklets*, short sequences of detections that can be safely connected to each other based on both appearance and space-time affinities. The second phase reasons over the entire temporal window, and partitions tracklets into identities (a.k.a. trajectories). Each phase has in turn a first stage that does a preliminary partitioning done safely by simple means in order to reduce the size of the BIP in that phase, and a second stage that solves a BIP exactly to utilize all evidence optimally. The four stages are now described in turn.

Space-Time Groups. The first stage divides the entire video sequence into 1-second intervals and uses hierarchical agglomeration [2] to group detections within each interval into *space-time groups* (Figure 1(b)). Initially, each detection is in a separate group. The algorithm then repeatedly merges the pair of groups that are closest to each other in space until k_i space-time groups are formed for time interval i. We set k_i to one half of the expected number of visible people in the given time interval, estimated as the ratio between the total number of detections and the number of frames in the interval. Because of the conservative choice of k_i, it is unlikely that observations that belong together end up in different groups. Even if they do, one person will end up split into different identities, and the trajectory stage, described later, has an opportunity to undo the split.

Tracklets. The second stage solves a BIP exactly for the observations of each space-time group, using the correlations (5) for evidence. The resulting partitions are called *tracklets*, and are at most one second long by construction. Solving exact BIPs on space-time groups ensures that both appearance and space-time evidence are used optimally within this short time horizon. Missing detections are recovered using interpolation or extrapolation and tracklets shorter than 0.2 seconds are discarded as false positives.

Appearance Groups. The third stage reasons in appearance space and groups tracklets from the entire temporal window into appearance groups that will be processed independently of each other in the fourth stage. Non-parametric methods for discovering appearance groups [56] are a good fit for this stage. However, we use k-means and set the number k of clusters manually for simplicity.

The wholesale splitting of identities across different appearance groups is an irrecoverable error. However, appearance grouping is again conservative, in that two observations are grouped whenever they are even just loosely similar. The main assumptions in this stage are that a person's appearance can have only short-lived variations (*e.g.*, partial occlusions or shadows) and that person appearance does not change suddenly and dramatically (*e.g.*, a person putting on a rain coat while hidden behind an obstacle). The conservative nature of this stage typically prevents identity-split errors, and a few incorrectly assigned observations can

be handled similarly to false positives and false negatives.
Trajectories. The last stage in the cascade solves a separate BIP (exactly) for all the tracklets in each appearance group and within the entire temporal window, again using both space-time consistency and appearance similarity as evidence. Missing tracklets for each trajectory are inferred using interpolation, and very short trajectories (shorter than 2 seconds) are discarded as false positives. The reduction in the size of the BIPs in the second and fourth stage of our cascade allows processing long temporal windows of data in real time.

2.2.7 Unlimited Time Horizon

Typical surveillance video streams are unbounded in length and require real-time, online processing. To turn the method described so far into an online algorithm we employ a sliding temporal window. The temporal extent of the window is set ahead of time—and depending on application—so that the observations in it can be processed in real time. Video frames stream in continuously, and an off-the-shelf person detector provides the needed detections. One-second-long tracklets are continuously formed by stages 1 and 2 of the cascade, and added to the input data. Once a window is processed completely as explained next, it is advanced by half its temporal extent.

All the tracklets that are at least partially contained in the first window are fed to the second phase of the cascade. Stages 3 and 4 form partial trajectories, and missing and spurious observations are handled as explained in Section 2.2.6. Partial trajectories are never undone, but they can be extended from data in subsequent windows. In windows after the first, the elementary input observations for stage 3 are all the tracklets and all the partial trajectories whose temporal extents overlap the current window. Except for this difference, the computations are the same as in the first window. This process repeats forever, and creates trajectories that an additional stage, based on a similar sliding window and one more binary integer program computation, partitions into identities.

2.3 Performance Measures

While theory guides descriptor and algorithm design, evidence of good performance must in the end be empirical. Current performance measures such as Multiple Object Tracking Accuracy (MOTA) [18] apply mainly to single-camera tracking. Even there, they suffer from various weaknesses [70, 61] that can be traced back to the *truth-to-result matching problem* defined in the introduction: Which of several computed identities that match part of a given ground-truth trajectory gets how much credit for its match?

Current measures tailored to multi-camera tracking emphasize inter-camera errors rather than within-camera ones, since they focus on *handover* errors [54]. Specifically, *fragments* count true identities that correspond to multiple computed ones, either across (X-FRG) or within (R-FRG) cameras, and *ID switches* count the reverse error, that is, multiple true identities falsely linked by the computation, either across (X-IDS) or within (R-IDS) cameras. These measures suffer from the truth-to-result matching problem as well.

A recent measure aimed at identity management [29], MCTA, combines true positive accuracy with resilience to identity switches within and between cameras. Identity switches between cameras are counted when there is an identity change between the last frame of the true trajectory in one camera and the first frame of the same trajectory in a different camera. This is clearly not the best mapping choice in case of trajectory fragmentation, where the correct identity might be correctly tracked for large part of the true trajectory and eventually lost in late frames. When a person returns to the same camera but the mapping has changed this measure in addition doesn't count this error as a between-camera ID switch but as a single camera ID switch.

To address these issues, we first solve the truth-to-result matching problem, and then build standard measures such as precision, recall, and F_1-score on top of that solution.

2.3.1 Matching True and Computed Identities

Our solution is based on the following remarks: (1) A correct match (no fragmentation or ID switches) between true identities and computed identities is one-to-one. (2) A fair truth-to-result matching is most favorable to the algorithm. (3) A fair penalty for an error of either type (ID switch or fragmentation) is the number of mis-assigned frames. Together, these considerations suggest computing a bipartite matching between computed and true identities with a minimal number of mis-assigned frames.

To this end, we construct a bipartite graph $G = (V_T, V_C, E)$ as follows. Vertex set V_T has one "regular" node τ for each true trajectory and one "false positive" node f^+_γ for each computed trajectory γ. Vertex set V_C has one "regular" node γ for each computed trajectory and one "false negative" node f^-_τ, for each true trajectory τ. Two regular nodes are connected with an edge $e \in E$ if they overlap in time. Every regular true node τ is connected to its corresponding f^-_τ, and every regular computed node γ is connected to its corresponding f^+_γ.

The cost on an edge $(\tau, \gamma) \in E$ tallies the number of false negative and false positive image frames that would be incurred if that match were chosen. Specifically, let $\tau(t)$ be the sequence of detections for true trajectory τ, one detection for each image frame t in the set $\mathcal{T}_\tau$, and define $\gamma(t)$ for $t \in \mathcal{T}_\gamma$ similarly for computed trajectories. The two simultaneous detections $\tau(t)$ and $\gamma(t)$ are a *mismatch* if they do not overlap in space, and we write

$$\mu(\tau, \gamma, t, \Delta) = 1 \; . \tag{17}$$

When both τ and γ are regular nodes, spatial overlap between two detections can be measured either in the image plane or on the reference ground plane in the world. In the first case, we declare a mismatch when the area of the intersection of the two detection boxes is less than $0 < \Delta < 1$ times the area of the union of the two boxes. On the ground plane, we declare a mismatch when the positions of the two detections are more than $\Delta = 1$ meter apart. If there is no mismatch, we write $\mu(\tau, \gamma, t, \Delta) = 0$. When either τ or γ is an irregular node (f^-_τ or f^+_γ), any detections in the other trajectory are mismatches. When both τ and γ are irregular, μ is undefined.

With this definition, the cost on edge $(\tau, \gamma) \in E$ is defined as follows:

$$c(\tau, \gamma, \Delta) = \underbrace{\sum_{t \in \mathcal{T}_\tau} \mu(\tau, \gamma, t, \Delta)}_{\text{False Negatives}} + \underbrace{\sum_{t \in \mathcal{T}_\gamma} \mu(\tau, \gamma, t, \Delta)}_{\text{False Positives}} \tag{18}$$

We define costs in terms of binary mismatches, rather than, say, Euclidean distances, so that a mismatch between regular positions has the same cost as a mismatch between a regular position and an irregular one. Matching two irregular trajectories incurs zero cost because they are empty.

A minimum-cost solution to this bipartite matching problem determines a one-to-one matching (consistently with remark 1 above) that minimizes the cumulative false positive and false negative errors (remark 2), and the overall cost is the number of mis-assigned frames (remark 3). Every (τ, γ) match is a True Positive (TP). Every (f^+_γ, γ) match is a False Positive (FP). Every (τ, f^-_τ) match is a False Negative (FN). Every (f^+_γ, f^-_τ) match is a True Negative (TN).

2.3.2 Performance Scores

We use the TP, FP, FN counts from bipartite matching to compute precision P and recall R:

$$P = \frac{TP}{TP + FP} = \frac{TP}{C} \quad , \quad R = \frac{TP}{TP + FN} = \frac{TP}{T} \tag{19}$$

where $C = TP + FP$ is the number of computed detections and $T = TP + FN$ is the number of true detections. Precision is the fraction of computed detections that are correct. Recall is the fraction of true detections that are computed. The F_1 score is a single figure that balances precision and recall and is the fraction of computed detections per average number of true and computed detections:

$$F_1 = 2\frac{PR}{P + R} = \frac{TP}{\frac{T+C}{2}} \; . \tag{20}$$

Precision, recall, and F_1-score are based on a mapping that is computed jointly over all trajectories by optimizing a single objective function and that treats within- and across-camera matches uniformly. Their definition is conceptually simpler than MOTA and MCTA, as it entails one global optimization rather than one optimization problem per frame. Precision and recall shed light on tracking trade-offs, while the F_1 score allows ranking all trackers on a single scale.

The bipartite matching figures also reveal the generalization cost incurred by solving the tracking problem simultaneously for all cameras rather than separately for each camera. Specifically, let

$$E_M = FP_M + FN_M \quad \text{and} \quad E_S = FP_S + FN_S \tag{21}$$

be the total number of errors for the multi-camera solution and for the solution obtained for separate cameras. Then,

$$E_M \geq E_S \tag{22}$$

because the multi-camera solution is feasible for separate cameras, but not necessarily *vice versa*. So the difference $E_M - E_S > 0$ can be interpreted as the fitting cost that has to be paid to make single-camera solutions hold across all cameras. Simple manipulation also shows that the F_1 score for the multi-camera solution is no greater than that for separate camera solutions is upper bounded by that for separate camera solutions:

$$F_1^M \leq F_1^S \; . \tag{23}$$

Our evaluation methodology addresses even the most glaring issues with MOTA and MCTA. For MOTA, consider a single-camera true trajectory with, say 16 frames, and suppose that the system fragments this identity into two, with eight consecutive frames each. Both MOTA and MCTA give a score of 93% to this outcome that seems half wrong. MCTA gives a correct score of 50% if the true trajectory is split in half over two cameras, but then it is puzzling that it scores what is effectively the same error so differently for different numbers of cameras. In contrast, the F_1 score based on our evaluation paradigm assigns 50% to all these cases—a more plausible and fully consistent score.

2.4 Experiments

Despite the large body of literature tackling identity management, the lack of public and realistic data sets, unpublished codes from other methods, and inconsistencies in the evaluation protocol make comparative performance evaluation difficult. In this Section, we first evaluate our methods on standard benchmark data and with standard performance evaluation measures to show that our methods improve on the state of the art in several aspects. These comparisons are made on single-camera data sets, to which most of the published results apply. After that, we describe our own, new multi-camera data set and propose preliminary results on it. We cannot compare these results directly with the literature, since our data set is not yet published. Instead, we make a few comparisons on smaller, existing data sets.

2.4.1 Single-Camera Experiments

We evaluate our algorithm on three standard single-camera data sets for multi-person tracking: PETS2009 [42], Town Center [15] and Parking Lot [1]. We used the PETS2009-S2L1 View 1 sequence, which has a resolution of 768 x 576 pixels and consists of 798 frames at 7 fps (117 seconds). The scene is not heavily crowded, but the low predictability in people's motion and a few long occlusions behind a lamp post makes the sequence challenging. The Town Center sequence is more challenging because it is longer, more crowded, and has longer occlusions. Occlusions in this sequence are mainly caused by people walking very close to each other. The sequence has a resolution of 1920 x 1080 pixels and consists of 4500 frames at 25 fps (180 seconds). The Parking Lot sequence consists of 998 frames at 30 fps (33.26 seconds) and has a resolution of 1920 x 1080 pixels. This sequence is challenging because it is filmed from an oblique angle and several people have similar appearance. Also, people walk close to each other in parallel causing long occlusions, both partial and full.

We use the standard Multiple Object Tracking Accuracy (MOTA) score [51] to evaluate the performance of our algorithm. This score combines the number of false positives $f_p(t)$, false negatives $f_n(t)$, and identity switches $id(t)$ over all frame indices t as follows:

$$MOTA = 1 - \frac{\sum_t (f_p(t) + f_n(t) + id(t))}{\sum_t g(t)} \tag{24}$$

where $g(t)$ is the ground-truth number of people in frame t. MOTA is widely accepted in the field as one of the principal indicators of a tracker's performance.

In Table 2 we present results for all sequences. We outperform state of the art methods in MOTA and identity switches. For a fair comparison, we use the detections used in previous work [1], courtesy of the authors. All evaluations are done using the CLEAR MOT evaluation script [5] and we use the standard 1 meter acceptance threshold.

In the PETS2009 sequence we use a long temporal window of 20 seconds and one appearance group since the scene is not crowded. We allow tracklets to be at most 10 frames in this sequence due to its low frame rate. The total running time of our method, not accounting for person detection, is 38 seconds. In the Town Center sequence we use a temporal window of 12 seconds and 5 appearance groups because the sequence is more crowded. Tracklets have lengths of at most 20 frames. The total running time on this sequence is 176 seconds, 120 of which were spent finding all tracklets. In the Parking Lot sequence we use a temporal window of 6 seconds and tracklets are at most 20 frames long. We used one appearance group in this sequence since it is short. The total running time on this sequence is 34 seconds.

	PETS2009	
	MOTA	IDsw
Berclaz [17]	80.00	28
Shitrit [12]	81.46	19
Andriyenko [3]	81.84	15
Henriques [46]	87.95	10
Izadinia [47]	90.70	-
Zamir [1]	91.50	8
Ours	**93.34**	**1**

	Town Center	
	MOTA	IDsw
Benfold [15]	64.9	259
Zhang [79]	65.7	114
Leal-Taixe [55]	67.3	86
Izadinia [47]	75.70	-
Zamir [1]	75.59	-
McLaughlin [60]	76.46	-
Ours	**78.43±0.29**	**68**

	Parking Lot	
	MOTA	IDsw
Izadinia [47]	88.90	-
Zamir [1]	92.27	1
Ours	**94.20**	**1**

Table 2: Multi Object Tracking Accuracy (MOTA) and ID switches on three standard data sets. MOTA variance for the Town Center sequence was caused by differences in appearance groups in different runs and resulting from the randomness of the k-means clustering algorithm.

Window Length, Accuracy, and Runtime Figures 3(a) and 3(b) show the dependency of tracking accuracy and running time on the length of the sliding window for the Town Center Sequence with 10 appearance groups. We ran several experiments on this sequence by progressively elongating the temporal window.

Figure 3(a) shows that after the temporal window length is increased beyond 3 seconds, which corresponds to the typical occlusion length in the scene, there is no significant improvement in the quality of the solution. The variations in the graph are caused by differences in the appearance groups that the k-means algorithm finds in each window. The slight decrease in the scores for windows longer than 19 seconds is because the parameter β in Equation (3) also influences how large partitions can grow in time.

Figure 3(b) shows how the sliding window length affects the running time. Appearance grouping allows us to achieve an unprecedented temporal window length for real-time computation.

Appearance Groups, Accuracy, and Runtime Figures 3(c) and 3(d) show the dependency of tracking accuracy and running time on the number of appearance groups for the Town Center sequence with a temporal window of 8 seconds.

Figure 3(c) shows that even a moderately high number of appearance groups, around 20, has negligible harmful effects on the accuracy of the tracker. When the number of appearance groups is increased further, the accuracy measure starts to decay because identities are split into separate groups. The fluctuations in the graph are again caused by the k-means algorithm, which over-clusters in windows that contain few tracklets.

Figure 3(d) shows that the overall running time is greatly reduced when we go from 1 to about 5 appearance groups, while the MOTA score only drops from 79% to 78.4% (Figure 3(c)). Increasing the number of appearance groups further yields marginal reductions in running time. The slight increase in total runtime for more than 20 appearance groups is caused by the k-means algorithm, whose complexity increases with k.

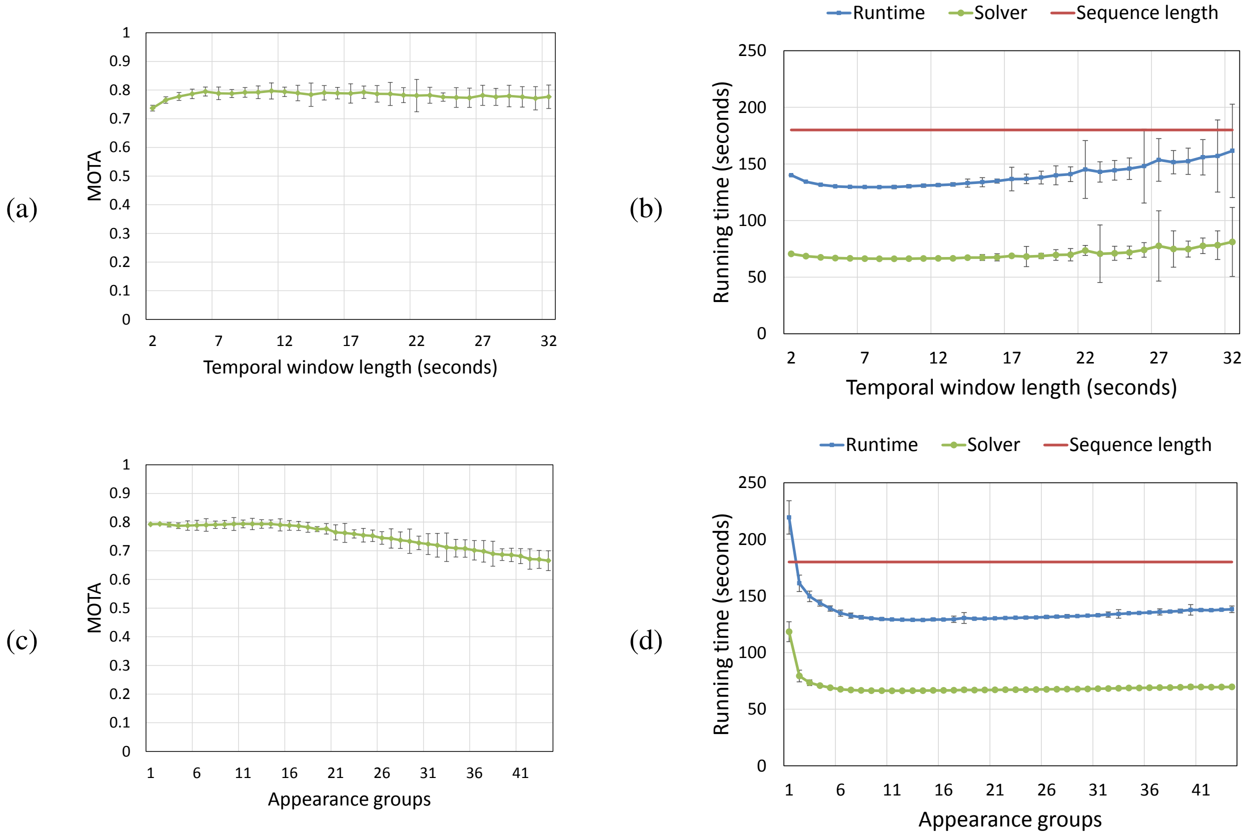


Figure 3: MOTA scores (a, c) and running times (b, d) as functions of the length of the sliding window (a, b) and the number of appearance groups (c, d) for the Town Center sequence. Solver time indicates how much time was spent for assembling and solving all the Binary Integer Programs. The total running time also includes the time for computing correlations, but does not account for person detection. Figures (a) and (b) are for ten appearance groups, and Figures (c) and (d) are for 8-second sliding windows. Best viewed on screen.

Approximate and Exact Graph Partitioning Solvers We explore the trade-off between accuracy and runtime for different combinations of solvers for graph partitioning. We demonstrate that approximating the solution of multi-person tracking by piecing together exact solutions of small subproblems is qualitatively better than algorithms with no optimality guarantees, while still achieving real-time performance.

Three algorithms for graph partitioning have been recently proposed in the literature [6], namely: Expand-and-Explore, Swap-and-Explore, and Adaptive Label Iterative Conditional Modes (AL-ICM). We use the latter in our experiment because of its speed and ability to scale to large problems. Given a labeling vector $L = \{1, 2, \dots\}^n$ the algorithm assigns a label l_u to observation u so as to minimize the following energy function:

$$E(L) = \sum_{uv} w_{uv} \mathbf{1}_{[l_u \neq l_v]} \tag{25}$$

where $\mathbf{1}_{[P]}$ is 1 when P is true and 0 otherwise. This energy is lowered when observations supported by negative correlation are labeled differently and when observations supported by positive correlation are

		PETS2009			Town Center			Parking Lot		
	Method	MOTA	Runtime	Solver	MOTA	Runtime	Solver	MOTA	Runtime	Solver
	Izadinia [47]	90.70	-	-	75.70	-	-	88.90	-	-
	Zamir [1]	91.50	-	-	75.59	-	-	92.27	-	-
Ours	**AL-ICM**	91.34	**9.33**	**0.31**	77.78 ± .35	**107.54**	**3.52**	93.33	**15.69**	**0.39**
	AL-ICM-NoGroup	92.20	10.68	0.45	78.46	284.73	18.86	93.92	28.00	1.11
	BIP	**93.18**	17.06	8.06	78.43 ± .29	177.17	73.10	**94.20**	33.59	20.40
	BIP-NoGroup	**93.18**	27.43	16.87	**78.87**	25725.82	23444.58	**94.20**	334.45	307.39

Table 3: Different combinations of solvers evaluated on three standard data sets. The length of each sequence is 117, 180 and 33.26 seconds respectively. Solver time indicates how many seconds were spent for solving graph partitioning problems in each sequence. The total running time also includes the time for computing correlations, but does not account for person detection.

labeled identically. This discrete energy minimization formulation has the advantage that the labeling vector L consists of n variables whereas the co-identity matrix X in our formulation consists of n^2 variables. This allows AL-ICM to scale to $n \geq 100,000$ observations.

AL-ICM is a greedy search algorithm. In each iteration, every variable is assigned the label that minimizes the energy, conditioned on the current label of the other variables. While ICM requires a fixed number of labels [19], AL-ICM handles a varying number of labels as follows: conditioned on the current labeling, each observation is assigned to the most rewarding partition, or to a new partition if penalized by all current partitions. The algorithm terminates either when the energy cannot be minimized further or when a predefined number of iterations is reached.

We construct two methods based on this algorithm. Method AL-ICM uses the greedy algorithm in stages 2 and 4 of the cascade, and space-time and appearance grouping in stages 1 and 3. Method AL-ICM-NoGroup uses the greedy algorithm but no grouping, thus only stages 2 and 4 of the full cascade.

We refer to our full algorithm as BIP, and we compare it also to a method we call BIP-NoGroup, that is, stages 2 and 4 of the cascade without space-time and appearance grouping. Performance metrics for all methods on three sequences are presented in Table 3.

Accuracy. All our methods consistently outperform the state of the art. Even method AL-ICM is on par, if not better than the state of the art, although it can be penalized by mistakes due to grouping heuristics and the suboptimal greedy algorithm. The differences in accuracy between our methods that use grouping and their corresponding version without grouping is minimal. This confirms that stages 1 and 3 of our cascade can be used in practice, safely and with negligible harmful effects. It is also worth noting that piecing together optimal solutions of small problems is superior to combining approximate solutions of small problems, which is common in the literature: Both BIP and BIP-NoGroup perform better than AL-ICM and AL-ICM-NoGroup, respectively.

Runtime. It is not surprising that the AL-ICM algorithm is much faster than the BIP solver. AL-ICM is a greedy algorithm and does not require assembling and solving a BIP with a quadratic number of variables and a combinatorial number of constraints. We note that the use of grouping heuristics is crucial for improving runtime performance; methods that do not use heuristics need to compute large and full correlation matrices. While the best time performance is that of AL-ICM, our BIP method is also fast enough to work in real-time at 25 fps.

Trade-offs. Considering the trade-offs between accuracy and runtime, the BIP approach is appropriate when accuracy is important and the scene has medium crowd density. The AL-ICM variant is more appropriate for time-critical applications or more crowded scenes, but comes with a cost in terms of accuracy. In

	MOTA ↑	MOTP ↑	PREC ↑	REC ↑	IDS ↓
KSP	65.56	64.70	84.26	80.89	146
Ours	70.10	60.50	89.02	80.46	281

Table 4: Comparison of our clique-based optimization approach against KSP's path-based formulation on the ISSIA data set. Arrows up (down) mean that greater (smaller) is better.

the absence of heuristics, which are not useful when the scene is crowded or all appearances look the same, AL-ICM-NoGroup is the only method from the above set that can be used to meet weaker time constraints.

2.4.2 Multi-Camera Experiments

A New Data Set Our new Duke Chapel data set is recorded with 8 video cameras pointed to a large outdoor area of the Duke University campus. Video is recorded at 1080p resolution and 60 frames per second. All cameras are calibrated and synchronized to within 1 second. Two camera pairs have small overlapping fields of view, while the fields of view of other cameras are disjoint. The data set is 1 hour 25 minutes long for each camera and has more than 1 million frames.

Exterior calibration information is available for each camera. Combined with manual annotation of every detection in every frame from every camera, this information allows providing ground truth in the form of a trajectory for each identity and camera on the world ground plane. Image bounding boxes for each person in every frame are also available and have been generated semi-automatically.

The data set includes 6791 trajectories for 2834 different identities (distinct people), for an average of 2.5 trajectories per identity. Some identities have up to 7 trajectories, meaning that the corresponding people appear and reappear up to 7 times as they walk around campus. The time period over which the cameras were on included intervals between scheduled classes, in which pedestrian traffic is heavy. As a result, there are between zero and 54 people per frame. The total number of hand-overs—transitions of the same person from one camera to the next—is 4159, while the maximum number of people simultaneously traversing blind spots is 50.

The first 5 minutes of video from each of the cameras have been set aside as a training and validation set and can be used to manually set or automatically estimate algorithm parameters.

Evaluation of our Trajectory Correlation Measure Our first multi-camera experiment evaluates the methods we described in Section 2.2.4 to compare trajectories. We compare our method to a widespread algorithm that uses K-Shortest Paths (KSP) [13, 16]. The experiment is conducted on the ISSIA data set [40], which is a 3 minutes soccer scene comprising 25 targets (11 from each team and 3 referees), recorded by 6 cameras with different levels of overlap between views and no blind spots. This setting let us test the ability of our method to take advantage of redundant data from overlapping views, as opposed to KSP which is specifically designed to work under the hypothesis of no blind spots. Moreover, to emphasize the merit of our trajectory correlation measure independently from how trajectories are described, we employ a simple HSV histogram (16, 4 and 4 bins respectively) to describe detections. For a fair comparison, both methods are run on the same input detections [41] and on the whole length of each sequence. KSP parameters have been set according to the authors' guidelines to obtain best results.

Results reported in Table 4 show that our method, despite not being specifically designed to work for overlapping views, is still able to exceed the state of the art in both accuracy and precision. KSP's lower number of identity switches and fragmentations is due to their additional entry/exit constraints: No soccer

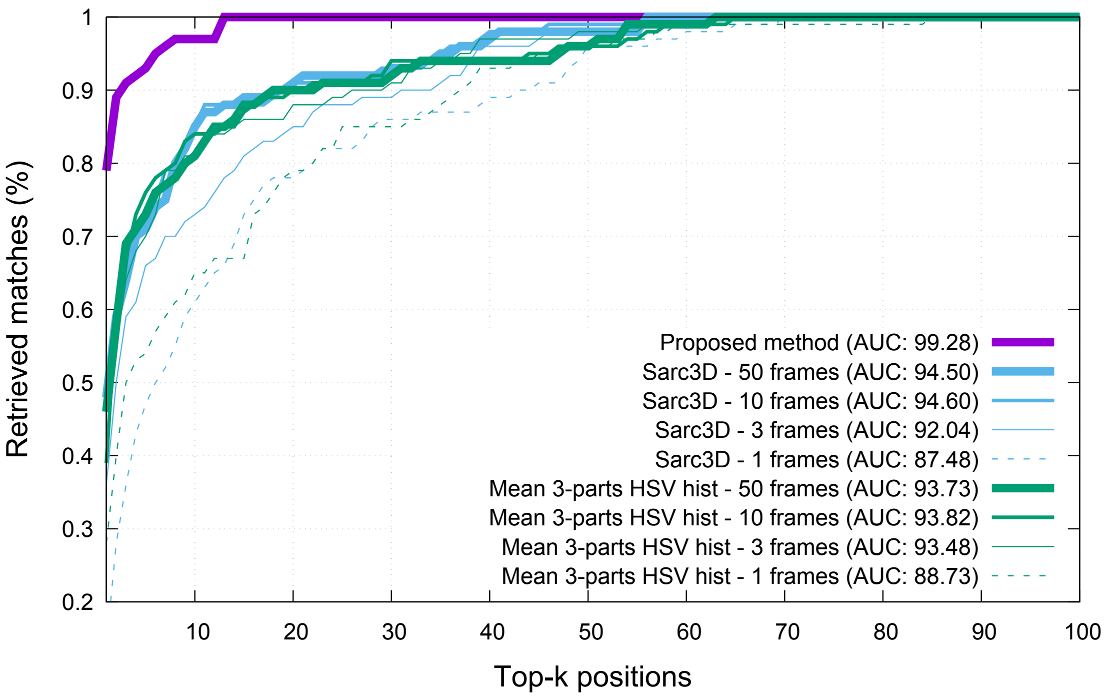


Figure 4: Retrieval results on a subset of our data set (Sec. 2.4.2) based on appearance only. Existing methods fail in exploiting the availability of more frames to increase the descriptor robustness.

player is allowed to enter or exit the field during the sequence, forcing trajectories to always exist. We do not take advantage of these constraints.

Conversely to the previous experiment, we now evaluate our trajectory descriptors in isolation, without global matching optimization. To this end, we extracted 100 pairs of trajectories, two for each of 100 identities, from our Duke Chapel data set. These trajectories come with bounding boxes in each frame, and are automatically extracted by employing the first two layers of our identity management pipeline. For a comparison, we reimplemented a previous method, Sarc3D [8], and tested it on our data. All the parameters in Sarc3D have been set to achieve best retrieval accuracy. This method shares some similarities with ours in that it addresses view invariance through the same 3D model and was designed to work with multiple (not necessarily continuous) snapshots. We also compare against a simple baseline descriptor, a histogram of HSV features (with 16, 4 and 4 bins in three different trials) of the pixels contained in the top, middle, and bottom third of each detection box, averaged over the frames of a trajectory.

The ordinate of the plot in Figure 4 shows the number of correct matches that are retrieved in the first top k ranked positions, where k is shown on the abscissa. Our method is retrieves 30% more correct matches as the top-ranked one when compared with other methods, and retrieves 90% of all correct matches in the top three positions. For others method we also run experiments by increasing the number of regularly sampled frames along the trajectory, as noted in the legend of the plot. After a few frames, performance saturates and then worsens over time, as a result of the lack of a proper way to aggregate and compare redundant descriptors. Sarc3D and the HSV baseline perform similarly to each other, with ours a clear winner.

A First Large-Scale Multi-Camera Experiment Table 5 summarizes standard performance figures, as well as our own new measures, on the Duke Chapel data set. This table is meant as a set of reference figures

	CLEAR MOT Metrics									Proposed Metrics		
	FP ↓	**FN** ↓	**IDS** ↓	**FRG** ↓	**MOTA** ↑	**MOTP** ↑	**GT**	**MT** ↑	**ML** ↓	**P** ↑	**R** ↑	**$\mathbf{F_1}$** ↑
Camera 1	9.70	52.90	178	366	37.36	67.57	1175	105	128	79.17	44.97	57.36
Camera 2	21.48	29.19	866	1929	49.17	61.70	1106	416	50	69.11	63.78	66.34
Camera 3	7.04	39.39	134	336	53.50	63.57	501	229	42	81.46	55.11	65.74
Camera 4	10.61	33.42	107	403	55.92	66.51	390	128	21	79.23	61.16	69.03
Camera 5	3.48	23.38	162	292	73.09	70.52	644	396	33	84.86	67.97	75.48
Camera 6	38.62	48.21	1426	3370	12.94	48.62	1043	207	91	48.35	43.71	45.91
Camera 7	8.28	29.57	296	675	62.03	60.73	678	373	53	85.23	67.08	75.07
Camera 8	1.29	61.69	270	365	36.98	69.07	1254	369	236	90.54	35.86	51.37
Over all cameras	14.38	43.85	3439	7736	41.66	63.54	6791	2223	654	72.25	50.96	59.77
Multi-Camera	Our method with proposed 3D descriptor									48.90	34.09	40.17
Multi-Camera	Our method with 2D HSV feature									48.61	33.88	39.93

Table 5: Single-camera (white rows) and multi-camera (grey rows) results on the Duke Chapel data set. For each camera we report both standard Multi-Target tracking measures as well as our new measure. FP, FN, P, R and F_1 are percentage values. Arrows up (down) mean that greater (smaller) is better.

to which future algorithms may be compared. No comparison is currently possible, as no other method, at the time of this writing, has been evaluated on data sets of this size and complexity.

Computation Time We implemented our algorithms in MATLAB and we used the Gurobi Optimizer to solve the Binary Integer Programs. All experiments were done on a PC with Intel i7-3610 2.3 GHz processor and 6 GB of memory. The results for the BIP-NoGroup method in Table 3 were produced on a Linux machine with Intel Xeon E5540 2.53 GHz processor and 96 GB memory in order to solve very large Binary Integer Programs.

Our approach takes on average 45 minutes per camera to compute trajectories for an 80-minute video, without counting person detection. It takes an additional 30 minutes to aggregate trajectories into identities. Since trajectories can be computed in parallel for different cameras, and person detectors can be run in real time on a dedicated GPU for each camera, our method achieves real-time performance. However, we have not built an end-to-end multi-camera system because we did not have the necessary computation hardware and programming manpower.

2.5 Conclusion

The identity management developed under this grant is the first end-to-end system that has shown to be able to track thousands of people from multiple cameras. Our method uses the same optimization framework at all levels and relies on a novel measure of trajectory similarity that aggregates information over time with reduced blurring. We created a new, large, fully annotated data set that we are about to make available to the research community, and we developed a new measure for performance evaluation that treats within-camera and across-camera errors uniformly and matches the complexity of the task. The plug-and-play nature of our system along with our new data set and evaluation methodology make past and future work easier to evaluate consistently.

While we solve each binary integer program optimally at each stage of the pipeline, we lose performance guarantees when the stages are combined. Our experiments show that we can achieve real-time identity management with good precision and recall performance. However, solving the entire problem optimally is an NP-hard problem that is still beyond the state of the art. We hope that our contributions will help keep pushing improvements of performance in this important area.

References

[1] A. D. Amir Roshan Zamir and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[2] M. R. Anderberg. Cluster analysis for applications. Technical report, DTIC Document, 1973.

[3] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272. IEEE, 2011.

[4] M. Ayazoglu, B. Li, C. Dicle, M. Sznaier, and O. Camps. Dynamic subspace-based coordinated multicamera tracking. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2462–2469, Nov. 2011.

[5] A. D. Bagdanov, A. Del Bimbo, F. Dini, G. Lisanti, and I. Masi. Posterity logging of imagery for video surveillance. 2012.

[6] S. Bagon and M. Galun. Large scale correlation clustering optimization. *arXiv preprint arXiv:1112.2903*, 2011.

[7] D. Baltieri, R. Vezzani, and R. Cucchiara. Learning articulated body models for people re-identification. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 557–560, New York, NY, USA, 2013. ACM.

[8] D. Baltieri, R. Vezzani, and R. Cucchiara. Mapping appearance descriptors on 3d body models for people re-identification. *International Journal of Computer Vision*, 111(3):345–364, 2015.

[9] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 238–247. IEEE, 2002.

[10] A. Bedagkar-Gala and S. Shah. Multiple person re-identification using part based spatio-temporal color appearance model. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1721–1728, Nov 2011.

[11] A. Bedagkar-Gala and S. K. Shah. Part-based spatio-temporal model for multi-person re-identification. *Pattern Recognition Letters*, 33(14):1908 – 1915, 2012. Novel Pattern Recognition-Based Methods for Re-identification in Biometric Context.

[12] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 137–144, 2011.

[13] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 137–144, Nov 2011.

[14] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In L. Agapito, M. M. Bronstein, and C. Rother, editors, *Computer Vision - ECCV 2014*

Workshops, volume 8926 of *Lecture Notes in Computer Science*, pages 613–627. Springer International Publishing, 2015.

[15] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.

[16] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[17] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, 2011.

[18] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, (246309):1–10, 2008.

[19] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[20] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004.

[21] M. Bredereck, X. Jiang, M. Korner, and J. Denzler. Data association for multi-object Tracking-by-Detection in multi-camera networks. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, Oct. 2012.

[22] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280. IEEE, 2011.

[23] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In *ECCV*, 2010.

[24] A. Buchanan and A. Fitzgibbon. Damped Newton algorithms for matrix factorization with missing data. In *CVPR*, 2005.

[25] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625, 2012.

[26] A. A. Butt and R. T. Collins. Multiple target tracking using frame triplets. In *Computer Vision–ACCV 2012*, pages 163–176. Springer, 2013.

[27] Y. Cai and G. Medioni. Exploring context information for inter-camera multiple target tracking. In *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 761–768, Mar. 2014.

[28] S. Calderara, R. Cucchiara, and A. Prati. Bayesian-competitive consistent labeling for people surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):354–360, Feb 2008.

[29] L. Cao, W. Chen, X. Chen, S. Zheng, and K. Huang. An equalised global graphical model-based approach for multi-camera object tracking. *CoRR*, abs/1502.03532, 2015.

[30] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic. On pairwise costs for network flow multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5537–5545, 2015.

[31] K.-W. Chen, C.-C. Lai, P.-J. Lee, C.-S. Chen, and Y.-P. Hung. Adaptive Learning for Target Tracking and True Linking Discovering Across Multiple Non-Overlapping Cameras. *IEEE Transactions on Multimedia*, 13(4):625–638, Aug. 2011.

[32] X. Chen, L. An, and B. Bhanu. Multitarget Tracking in Nonoverlapping Cameras Using a Reference Set. *IEEE Sensors Journal*, 15(5):2692–2704, May 2015.

[33] X. Chen, K. Huang, and T. Tan. Direction-based stochastic matching for pedestrian recognition in non-overlapping cameras. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 2065–2068, Sept. 2011.

[34] D. Cheng and M. Cristani. Person re-identification by articulated appearance matching. In S. Gong, M. Cristani, S. Yan, and C. C. Loy, editors, *Person Re-Identification*, Advances in Computer Vision and Pattern Recognition, pages 139–160. Springer London, 2014.

[35] D. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *Proceedings of the British Machine Vision Conference*, pages 68.1–68.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.68.

[36] R. T. Collins. Multitarget data association with higher-order motion models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1744–1751. IEEE, 2012.

[37] S. Daliyot and N. S. Netanyahu. A Framework for Inter-camera Association of Multi-target Trajectories by Invariant Target Models. In J.-I. Park and J. Kim, editors, *Computer Vision - ACCV 2012 Workshops*, number 7729 in Lecture Notes in Computer Science, pages 372–386. Springer Berlin Heidelberg, 2013.

[38] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury. Consistent re-identification in a camera network. In *Computer Vision–ECCV 2014*, pages 330–345. Springer, 2014.

[39] A. Dehghan, S. M. Assari, and M. Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *CVPR*, volume 1, page 2, 2015.

[40] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 559–564. IEEE, 2009.

[41] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, Sept 2010.

[42] J. Ferryman. Proceedings (pets 2009). In J. Ferryman, editor, *Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2009)*. IEEE, 2009.

[43] A. Gilbert and R. Bowden. Tracking Objects Across Cameras by Incrementally Learning Inter-camera Colour Calibration and Patterns of Activity. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, number 3952 in Lecture Notes in Computer Science, pages 125–136. Springer Berlin Heidelberg, 2006.

[44] R. Hamid, R. Kumar, M. Grundmann, K. Kim, I. Essa, and J. Hodgins. Player localization using multiple static cameras for sports visualization. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 731–738, June 2010.

[45] P. Heckbert. Color image quantization for frame buffer display. In *Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '82, pages 297–307, New York, NY, USA, 1982. ACM.

[46] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2470–2477. IEEE, 2011.

[47] H. Izadinia, I. Saleemi, W. Li, and M. Shah. (mp)2t: Multiple people multiple parts tracker. In A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV (6)*, volume 7577 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2012.

[48] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146–162, Feb. 2008.

[49] W. Jiuqing and L. Li. Distributed optimization for global data association in non-overlapping camera networks. In *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–7, Oct. 2013.

[50] A. Kamal, J. Farrell, and A. Roy-Chowdhury. Information Consensus for Distributed Multi-target Tracking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2410, June 2013.

[51] B. Keni and S. Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[52] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[53] R. Kumar, G. Charpiat, and M. Thonnat. Multiple object tracking by efficient graph partitioning. In *Computer Vision–ACCV 2014*, pages 445–460. Springer, 2014.

[54] C.-H. Kuo, C. Huang, and R. Nevatia. Inter-camera Association of Multi-target Tracks by On-Line Learned Appearance Affinity Models. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, number 6311 in Lecture Notes in Computer Science, pages 383–396. Springer Berlin Heidelberg, 2010.

[55] L. Leal-Taixé, G. Pons-Moll, and B. Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 120–127. IEEE, 2011.

[56] C. Liu, S. Gong, C. C. Loy, and X. Lin. Person re-identification: What features are important? In *ECCV Workshops (1)*, pages 391–401, 2012.

[57] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[58] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*, volume 2, June 2004.

[59] N. Martinel, C. Micheloni, and G. L. Foresti. Saliency weighted features for person re-identification. In *Computer Vision-ECCV 2014 Workshops*, pages 191–208. Springer International Publishing, 2014.

[60] N. McLaughlin, J. M. del Rincon, and P. Miller. *Online Multiperson Tracking With Occlusion Reasoning and Unsupervised Track Motion Model.* 2013.

[61] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 735–742. IEEE, 2013.

[62] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, 2nd edition, 2006.

[63] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[64] S. Ricco and C. Tomasi. Simultaneous Compaction and Factorization of Sparse Image Motion Matrices. In *ECCV*, 2012.

[65] E. Ristani and C. Tomasi. Tracking multiple people online and in real time. In *Asian Conference on Computer Vision*, November 2014.

[66] E. Ristani and C. Tomasi. Tracking multiple people online and in real time. In *ACCV-12th Asian Conference on Computer Vision*. Springer, 2014.

[67] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[68] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):51–65, 2005.

[69] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1815–1821. IEEE, 2012.

[70] F. Solera, S. Calderara, and R. Cucchiara. Towards the evaluation of reproducible robustness in tracking-by-detection. In *Advanced Video and Signal Based Surveillance (AVSS), IEEE International Conference on*, Aug 2015.

[71] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, 2010.

[72] C. Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 526–527. Society for Industrial and Applied Mathematics, 2004.

[73] J. Tan. A note on the inapproximability of correlation clustering. *Information Processing Letters*, 108(5):331–335, 2008.

[74] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[75] B. L. Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29:350–362, 1938.

[76] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1282–1289. IEEE, 2014.

[77] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[78] J. Yao and J. Odobez. Multi-layer background subtraction based on color and texture. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[79] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[80] S. Zhang, E. Staudt, T. Faltemier, and A. Roy-Chowdhury. A Camera Network Tracking (CamNeT) Dataset and Performance Baseline. In *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 365–372, Jan. 2015.

[81] S. Zhang, Y. Zhu, and A. Roy-Chowdhury. Tracking multiple interacting targets in a camera network. *Computer Vision and Image Understanding*, 134:64–73, May 2015.

[82] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

A Dense, Long-Term Visual Tracking

The goal of long-range, high-density motion estimation in video analysis is to compute the life of every point in a dense sampling of the visible surfaces in the scene. The image projection of a scene point moves along a *path* in the image plane. Sometimes the point is visible, and sometimes it is occluded by some object in the world or by the boundaries of the image. In a *dense* motion estimate, at least one path passes through every pixel of the sequence.

Dense, long-range motion estimation supports a number of applications. The computed paths can propagate to multiple frames any annotations or edits made in a single frame, thereby easing video labeling and editing. If visible paths can be extrapolated into regions where they are occluded, the occluding object can be removed from the video by painting the pixels it occupies with the extrapolated colors. Videos can be segmented into separate objects by clustering paths into coherent groups. The shapes and appearance of the resulting tube-like regions can support the detection and recognition of objects and activities.

Image motion information is either poor or altogether unavailable where the scene has little or no visual texture—the so-called *aperture problem*. As a consequence, regularization—or priors in probabilistic parlance—must be employed to extrapolate motion information from textured to poorly textured regions. To this end, we assume that (i) image paths live in a low-dimensional space, (ii) appearance remains approximately constant along the visible portion of a path, and (iii) exactly one world point is visible at every image point. The first assumption is exactly satisfied with rigid motion, and approximately satisfied in many circumstances. The second assumption is pervasive in motion analysis, and the third excludes semi-transparent objects.

Model

Let p be an index into a set of *paths* $\mathbf{x}_p(t) : \mathcal{T} \to \mathbb{R}^2$, where $\mathcal{T}$ is the (discrete) time domain of the video sequence. A path is visible at time t iff its *visibility flag* $\nu_p(t) : \mathcal{T} \to \{0, 1\}$ is equal to 1 at time t. Both functions $\mathbf{x}_p(t)$ and $\nu_p(t)$ are unknowns to be estimated for all paths in a given video sequence. To ensure approximately (at first, and exactly later) at least one path per pixel in every frame, we *anchor* $\mathbf{x}_p(t)$ to point $\mathbf{u}_p$ in some frame τ_p by letting $\mathbf{x}_p(\tau_p) = \mathbf{u}_p$, and require enough anchor points to have some path pass through every pixel in the video sequence. In contrast with LME, τ_p is path-specific and unrestricted.

Paths are assumed to be in the space spanned by a sequence-specific *basis* of paths $\{\varphi_1, \ldots, \varphi_K\}$, up to a shift:

$$\mathbf{x}_p(t) = \mathbf{u}_p + \sum_{k=1}^{K} c_{pk}(\varphi_k(t) - \varphi_k(\tau_p)) \; . \tag{26}$$

The motion relative to the anchor point $\mathbf{x}_p(\tau_p) = \mathbf{u}_p$ is determined by the unknown coefficients $\mathbf{c}_p = (c_{p1}, \ldots, c_{pK})$.

Since paths in a video with F frames have F points, the standard basis over $\mathbf{R}^{2F}$ can represent any path exactly. However, for many sequences a much more compact ($K << 2F$) basis is adequate, and provides powerful, sequence-specific regularization.

The model in equation (26) is Lagrangian, in the sense that it models individual points as they move through a video sequence. This formulation is in contrast with the more traditional Eulerian specification of optical flow, in which partial differential equations describe the flow that passes through fixed locations in the image. To rephrase, the observer is fixed in space in the Eulerian formulation, and moves with the flow in the Lagrangian one.

Given basis paths and anchor points, we find paths and visibility flags by interleaving computing optimal paths given visibility with computing optimal visibility given paths. The next two sections define the optimality criteria for these computations. The Sections thereafter show how to find the path basis and initial anchors, and how to compute optimal paths, visibility, and anchors.

Optimal paths Given a set of basis paths and a set of anchors, we find the best motion coefficients for each path by minimizing an objective function that penalizes changes in appearance along a path (temporal smoothness) and differences between nearby paths (spatial smoothness):

$$\sum_{p \in \mathcal{P}} \sum_{t=1}^{F} E_D(\mathbf{c}_p, t) + \lambda \sum_{p,q \in \mathcal{P}} E_S(\mathbf{c}_p, \mathbf{c}_q) \ . \tag{27}$$

The first term,

$$E_D(\mathbf{c}_p, t) = \nu_p(t) \Psi(I(\mathbf{c}_p, t) - I(\mathbf{c}_p, \tau_p)) \ , \tag{28}$$

employs a robust penalty function $\Psi(s) = \sqrt{s^2 + \epsilon^2}$ to measure the difference between the image intensity $I(\mathbf{c}_p, t) = I(\mathbf{x}_p(t))$ of the path in frame t and that at the anchor $\mathbf{u}_p$ in frame τ_p. Multiplication by $\nu_p(t)$ ensures that this penalty is levied only on visible points. The second term,

$$E_S(\mathbf{c}_p, \mathbf{c}_q) = \alpha_{pq} \sum_{k=1}^{K} \Psi(c_{pk} - c_{qk}) \ , \tag{29}$$

measures the difference between the motion coefficients of pairs of paths. The multiplier α_{pq} couples nearby paths that have similar appearance, and is equal to

$$\alpha_{pq} = \exp\left(-\frac{(I(\mathbf{c}_p, \tau_p) - I(\mathbf{c}_q, \tau_q))^2}{\sigma^2} \right) \tag{30}$$

if the path p is visible in the anchor frame of path q (that is, if $\nu_p(\tau_q) = 1$) and passes close enough to the anchor of q (that is, if $||\mathbf{x}_p(\tau_q) - \mathbf{u}_q|| < \Delta$). Otherwise, $\alpha_{pq} = 0$.

Optimal visibility The binary visibility flag $\nu_p(t)$ for each path and frame is modeled as a MRF whose structure depends on the current estimates $\mathbf{x}_p(t)$ of the paths $p \in \mathcal{P}$. The MRF has one node for each point $\mathbf{v}_p(t) = (\mathbf{x}_p(t), t)$ along some path, for $t = 1, \ldots, F$, and one binary random variable $\nu_p(t)$ per node. The neighborhood of $\mathbf{v}_p(t)$ is the set of points $\mathbf{v}_q(t)$ with $q \neq p$ and $\|\mathbf{v}_p(t) - \mathbf{v}_q(t)\| \leq \Delta$ for some small fixed Δ (spatial neighborhood), plus the two points $\mathbf{v}_p(t-1)$ and $\mathbf{v}_p(t+1)$ that are temporally adjacent to $\mathbf{v}_p(t)$ along path p (temporal neighborhood).

Each node in the MRF is associated with a binary *observed visibility* flag $\hat{\nu}_p(t)$ computed from the data as follows. Path points in each frame are scored by their *consistency*, which measures how little a patch around $\mathbf{v}_p(t)$ changes as it is transported by the current estimates of paths near $\mathbf{v}_p(t)$ to (i) a few frames before and after time t, and (ii) the anchor frame τ_p for path p, similar to LME. The *controlling path* at $\mathbf{v}_p(t)$ is the most consistent path through the spatial neighborhood of $\mathbf{v}_p(t)$. Let now

$$\bar{d}_{pq} = \frac{1}{F} \sum_{t=1}^{F} ||\mathbf{x}_p(t) - \mathbf{x}_q(t)|| \tag{31}$$

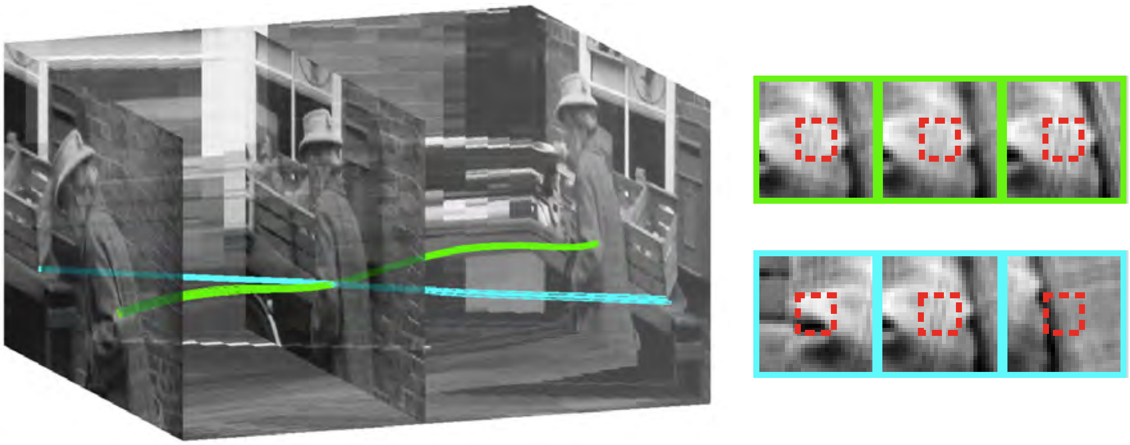


Figure 5: A spatiotemporal cube of the *marple7* sequence. Time runs from left to right. The corner of the crate (cyan) is first occluded by Miss Marple's arm (green) in frame 12. A small patch (red dashed squares) around each path in every frame is transported along the current path estimates and monitored for consistent appearance. The arm patch (top right) is most consistent, and makes this the controlling path at that point and frame. Points along paths that either coincide with or are substantially parallel to a nearby controlling path have their observed visibility flag $\hat{\nu}_p(t)$ set to 1. All other flags are set to 0. Observed flags affect the estimated visibility flags at the nodes of a MRF that enforces spatial and temporal consistency of the flags and ensures that at least one path is visible at every pixel.

be the average distance between two paths, and let p^* be the controlling path at $\mathbf{v}_p(t)$. Then, the observed visibility $\hat{\nu}_p(t)$ is defined as follows (see also Figure 5):

$$\hat{\nu}_p(t) = \begin{cases} 1 & \text{if } \bar{d}_{pp^*} \leq 4 \text{ pixels} \\ 0 & \text{otherwise} \end{cases} . \tag{32}$$

In words, a path p is observed to be visible at $\mathbf{v}_p(t)$ when it either coincides with ($p = p^*$ so that $\bar{d}_{pp^*} = 0$) or is nearly parallel ($\bar{d}_{pp^*} \leq 4$) to the controlling path p^* at $\mathbf{v}_p(t)$.

The observed visibility flags $\hat{\nu}_p(t)$ influence the (hidden) visibility flags $\nu_p(t)$ through a data term in the MRF. Let

$$\Delta I_p(t) = \Psi(I(\mathbf{c}_p, t) - I(\mathbf{c}_p, \tau_p)) \tag{33}$$

be the same per-path, per-frame measure of intensity consistency used in (28). We define the following average measure of intensity change along the visible portion of path p:

$$\Delta_p = \frac{\sum_{t=1}^{T} \hat{\nu}_p(t) \Delta I_p(t)}{\sum_{t=1}^{T} \hat{\nu}_p(t)} . \tag{34}$$

For correctly estimated paths, this measure reflects variations of intensity caused by unmodeled effects such as image noise or global illumination changes, rather than by occlusions. Given these definitions, the data term of the MRF is defined as follows:

$$\begin{aligned} D(\nu_p(t) = 1) &= \Delta I_p(t) + \lambda_L(1 - \hat{\nu}_p(t)) \\ D(\nu_p(t) = 0) &= \Delta_p + \lambda_L \hat{\nu}_p(t) . \end{aligned} \tag{35}$$

The terms with multiplier λ_L bias estimated visibility values $\nu_p(t)$ toward observed values $\hat{\nu}_p(t)$. Setting a point to be visible incurs the additional charge $\Delta I_p(t)$, equal to the change in intensity between anchor and current point. Setting a point to be invisible incurs the additional charge Δ_p that accounts for the fact that intensity variations may be caused by other than occlusions.

The weights on edges between the random variables of the MRF encourage both temporal and spatial consistency among visibility values. Specifically, a penalty

$$V\Big(\nu_p(t),\, \nu_p(t+1)\Big) = \lambda_T |\nu_p(t) - \nu_p(t+1)| \tag{36}$$

is added between temporally adjacent neighbors to discourage changes of visibility along a path. The weight on an edge between spatial neighbors is

$$V\Big(\nu_p(t),\; \nu_q(t)\Big) = \lambda_S w_{pq}(t) |\nu_p(t) - \nu_q(t)| \tag{37}$$

with

$$w_{pq}(t) = \frac{e^{-\left(\frac{\Delta I_{pq}(t) + \Delta I_{pq}}{\sigma^2}\right)}}{\bar{d}_{pq} + \epsilon} \tag{38}$$

where $\epsilon > 0$ prevents division by zero. In this expression,

$$\begin{aligned} \Delta I_{pq}(t) &= (I(\mathbf{c}_p, t) - I(\mathbf{c}_q, t))^2 \\ \Delta I_{pq} &= (I(\mathbf{c}_p, \tau_p) - I(\mathbf{c}_q, \tau_q))^2 \ . \end{aligned} \tag{39}$$

In words, $\Delta I_{pq}(t)$ measures difference in appearance between paths in a single frame, and ΔI_{pq} measures a similar difference between anchor points. The combined effect of these two terms is to push discontinuities in visibility closer to intensity boundaries, and the division by $\bar{d}_{pq}$ reduces the spatial discontinuity penalty between unrelated paths.

Finally, we clamp enough visibility values to 1 to ensure that every pixel in the sequence has a visible path through it. Specifically, we make all anchor points visible, $\nu_p(\tau_p) = 1$, and we also force $\nu_p(t) = 1$ if $\bar{d}_{pp^*} < \sqrt{2}$. This assignment guarantees that at least one visible path goes through every pixel because $\bar{d}_{p^*p^*} = 0$. We roll the pairwise cost for each edge incident to a clamped node into the unary cost for the other node of that edge.

Computation Preliminaries

Before we solve for motion and visibility, we select basis paths and an initial set of anchors, paths, and visibility flags as follows.

Finding the basis paths Basis paths are obtained by first tracking a sparse set of feature points with a frame-to-frame tracker [57]. This yields several *tracks*, that is, paths that do not necessarily extend through the entire sequence. These tracks are supplemented with those formed by concatenating optical flow vectors between consecutive frames [71], as described in more detail under *Initialization* below, where we do the same to initialize a dense set of paths.

For some sequences, several tracks may extend from first to last frame. PCA can then yield a basis whose size K is determined by adding principal components until the reconstruction residual for the input tracks is below, *e.g.*, 2 pixels.

In general, however, occlusions and tracking failures make tracks start late and end early, leading to a matrix of track coordinates with missing entries. We iterate between matrix factorization with missing data [24] and a compaction step that associates tracks corresponding to the same world point [64]. If needed, a user can be asked to correct mistakes in data association. We scale path coordinates so that the mean per-path motion between frames is one pixel.

Initialization To cover every pixel in a video sequence with paths, we need to create a number of paths of the same order of the number of visible points in the sequence. To this end, we form an initial set of paths by placing anchor points at every pixel in the first and last frames in the sequence, and supplement these with additional anchors in regions that are not yet covered by some path. More specifically, we start by defining path fragments we call *temporal superpixels* with the procedure described by Sundaram *et al.* [71]. We first concatenate optical vectors into tracks, which we break when the optical flow field fails a forward-backward consistency check or when the point is too close to a motion boundary (equations (5) and (6) from [71], respectively). To prevent merging foreground and background tracks, we create a thin empty buffer around the regions where tracks terminate. If a superpixel thus created extends over several frames, we replace it by an entire path whose coefficients are computed by projecting the superpixel onto the path basis. If the superpixel is too short, we copy the coefficients of the path with the greatest intensity consistency over a few frames among existing, nearby, parallel paths. The temporal extent of superpixels provides an initial estimate for the visibility flags.

Every temporal superpixel that extends to the first or the last frame yields an initial path anchored in that frame. The remaining superpixels are said to be *covered* if their paths differ by an average of less than two pixels per frame from an existing path. New paths are formed only for superpixels that are not yet covered, and their anchor points are placed in the last frame of the superpixel. Figure 6a shows the anchor points selected in this way for the *marple7* sequence. Colors other than gray are anchors, and similar colors correspond to similar sets of path coefficients.

After this initialization stage, the energy functions defined earlier are minimized by the algorithms described in the previous Section. This can result in the insertion of additional anchor points. Figure 6b shows the color-coded anchor points after convergence.

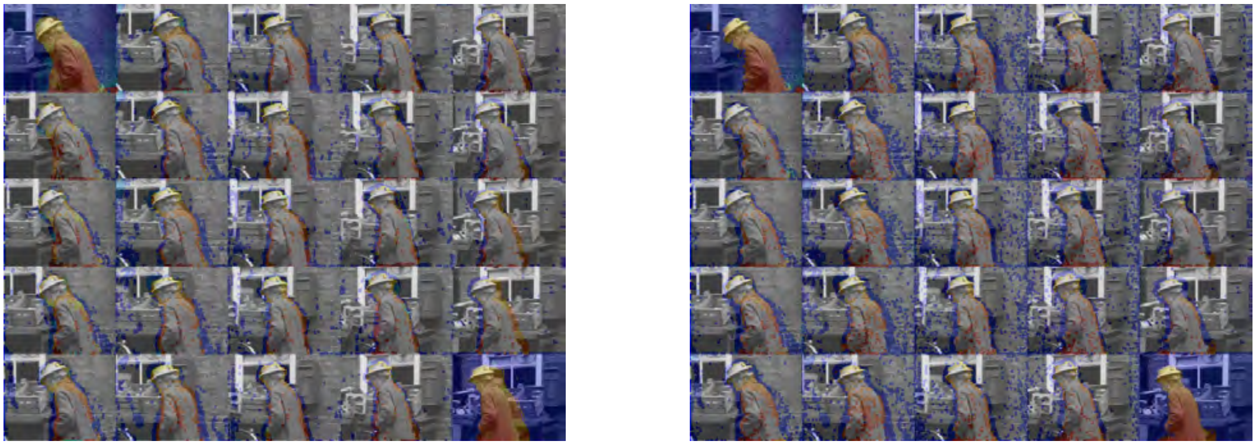
Optimization

Starting with the paths and visibility flags constructed as described above, we interleave two steps during optimization: a combinatorial optimization step finds visibility flags $\nu_p(t)$ for the current path estimates, and a continuous optimization step updates path coefficients $\mathbf{c}_p$ given the current visibility estimates. In the process, we add anchor points until every pixel in the sequence has at least one path through it, and remove anchors of invisible paths. We stop when the maximum change in every path falls below one pixel in every frame.

The initial path estimates are often poor along occlusion boundaries, because visibility is not yet accounted for. Because of this, we heuristically regroup paths between each combinatorial and continuous step to let foreground and background vie for paths between them.

We now describe the continuous step, path regrouping, combinatorial step, anchor management, and termination.

Continuous step. We update path coefficients by minimizing the energy function (27) via trust-region Newton Conjugate Gradients optimization [62]. This method only requires computing vectors of the form $H\mathbf{v}$ where H is the Hessian, rather than the very large but sparse H itself. The sparsity pattern of H changes



(a)

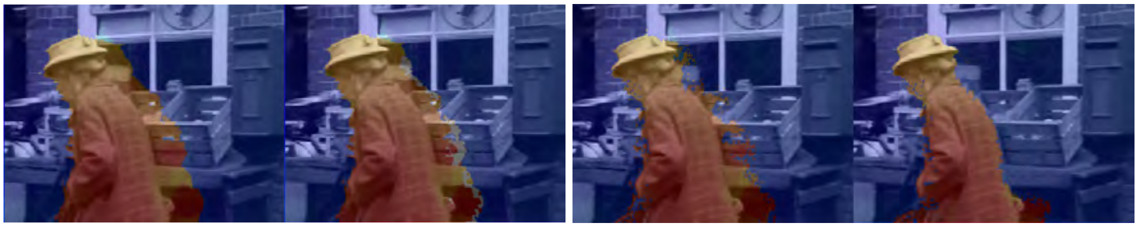
(b)

Figure 6: Anchor point selected during initialization (a) and at convergence (b). Colors other than gray denote anchor points, and similar colors denote similar sets of path coefficients. Note the improved segmentation of Miss Marple after convergence.

over time because the coupling coefficients α_{pq} in equation (30) depend in turn on the path coefficients. When computing successive conjugate gradients, we treat the terms α_{pq} as constants—a good approximation for small path perturbations—and recompute them between full descent steps.

Path regrouping. After 40 descent steps, we allow paths to copy their coefficients and visibility flags from one of their neighbors if doing so improves the path's fit to data. Specifically, path p copies from path q if $\nu_q(\tau_p) = 1$, $\tau_p \neq \tau_q$, $d_{pq}(\tau_p) < \Delta$, path q is visible for at least half the frames, and the copy improves the data fit for p the most. Figure 7 illustrates the benefits of this step on the *marple7* sequence.

Combinatorial step. Visibility flags are updated after path regrouping by using graph cuts [20, 52] to compute the MAP estimate for the MRF defined earlier. The energy function is amenable to this method as



(a) Without regrouping.

(b) With regrouping.

Figure 7: Effect of path regrouping. Motion estimates are shown using the same color scheme as in Figure 6. The first image in each pair shows the solution after the first round of optimization; the second shows results at convergence. Regrouping (b) recovers from a poor local optimum with incorrect estimates for the motion of the occluded background.

Sequence	Method	APIE	Path length (frames)		Path density (pixels)			% pixels containing visible paths
			Mean	Std. dev.	50th	95th	99th	
Flowerbed	LDOF traj.	13.97	11.2	10.5	0.47	8.5	15.2	66.6%
	LME	9.37	**23.9**	7.3	0.31	0.79	1.3	97.5%
	Ours	**6.10**	23.3	7.2	**0.29**	**0.66**	**0.84**	**99.8%**
Truck	LDOF traj.	19.74	6.8	7.5	1.2	47.0	70.0	47.3%
	LME	17.82	**23.4**	7.4	0.39	1.9	4.4	88.6%
	Ours	**9.80**	22.0	9.0	**0.27**	**0.65**	**0.86**	**99.8%**
Marple7	LDOF traj.	7.84	6.7	6.4	0.47	6.9	13.3	69.0%
	LME	6.28	**15.9**	7.5	0.43	5.7	9.7	76.0%
	Ours	**5.61**	15.5	6.8	**0.32**	**0.70**	**0.89**	**99.7%**
Marple1	LDOF traj.	15.83	9.4	11.4	0.51	14.7	26.7	62.5 %
	Ours	**8.79**	**18.9**	19.3	**0.35**	**0.85**	**1.0**	**97.8%**
Marple8	LDOF traj.	13.69	14.9	14.7	0.47	7.4	16.4	71.2%
	Ours	**9.30**	**45.8**	20.8	**0.25**	**0.65**	**0.87**	**99.7%**

Table 6: Solution quality metrics. APIE measures average intensity constancy along estimated paths (smaller is better, assuming the brightness constancy assumption holds). Path length is the number of frames in which a path is reported as visible (longer is typically better). Path density is computed by measuring the distance to the nearest visible path for each pixel. We report the 50th, 95th, and 99th percentiles (smaller is better), as well as the percentage of pixels with a visible path within a radius of 1 pixel (larger is better). The *marple1* and *marple8* sequences were too large for LME to complete within a reasonable timeframe. We stopped computation after 72 hours when only a single iteration had completed.

the edge costs (37) satisfy

$$\begin{aligned} V(0,0) + V(1,1) &\leq V(0,1) + V(1,0) \\ 0 &\leq V(0,1) + V(1,0) \; . \end{aligned} \tag{40}$$

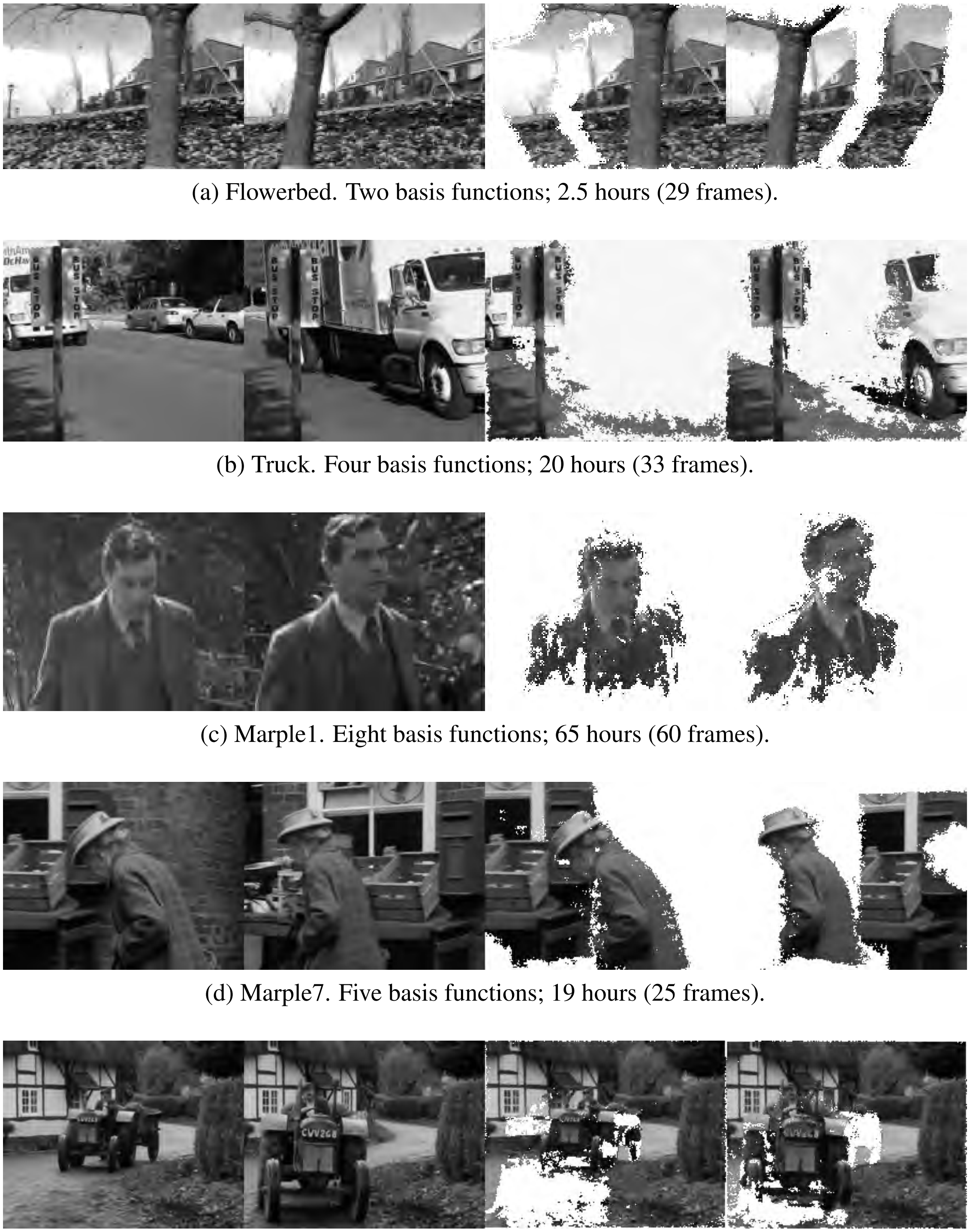
Anchor management. When the maximum change in any path in any frame is less than one pixel, we check that every pixel in the video has a visible path through it. If not, we add new anchor points to fill voids and resume optimization. Newly inserted paths copy their initial parameters from the closest visible path.

Anchors on paths that are invisible everywhere except at the anchor itself (which is always visible) are deleted. These one-point paths occur when visibility estimation correctly identifies an outlier with an incorrect path estimate.

Termination. Optimization terminates when all path estimates change by less than a pixel in every frame and all pixels in the video have a path through them.

Results

We evaluate the performance of our technique on five real sequences of increasing complexity, all with large motions and significant occlusions. The popular *flowerbed* (29 frames) and a new sequence with a *truck* driving behind a road sign (33 frames) contain only rigid motion. The three with non-rigid motion are from the Berkeley motion segmentation dataset [23]: 60 frames from *marple1*, 72 frames from *marple8*, and 25 frames from *marple7*. The *marple7* and *flowerbed* sequences are the same as those evaluated in LME. Figure 8 shows sample frames.



(a) Flowerbed. Two basis functions; 2.5 hours (29 frames).

(b) Truck. Four basis functions; 20 hours (33 frames).

(c) Marple1. Eight basis functions; 65 hours (60 frames).

(d) Marple7. Five basis functions; 19 hours (25 frames).

(e) Marple8. Eight basis functions; 68 hours (72 frames).

Figure 8: Results of our method. For each sequence, we show the first and last frames, followed by the last frame warped to align with the first frame, and vice versa. Regions detected as occluded in the source frame of the warp are marked in white. Solution times (rounded to nearest half-hour) exclude basis computation.

Qualitative evaluation For a qualitative evaluation, we use our motion results to warp all frames to a selected frame. This creates a motion-compensated video that should appear static except for regions that are occluded in a particular frame, and that we paint white. Figure 8 shows the last frame aligned to the first frame, and viceversa, for all sequences.

Quantitative evaluation It is difficult to get reliable ground truth paths for realistic sequences. Synthetic datasets [25] do not preserve associations across occlusions. Manual labeling for real sequences is painstaking and unreliable, particularly for complex motions or low-texture regions.

Instead, we measure the degree to which intensities remains constant along computed paths as a proxy for performance. We define the *all-path interpolation error* (APIE)

$$\text{APIE} = \sqrt{\frac{1}{N}\sum_{p}\frac{\sum_t \nu_p(t)(I(\mathbf{c}_p,t) - I(\mathbf{c}_p,\tau_p))^2}{\sum_t \nu_p(t)}} \tag{41}$$

where N is the number of paths. Even on perfect paths, APIE would measure the correctness of the brightness constancy assumption, and would be nonzero in general.

Table 6 reports the APIE for different methods for each sequence, computed with intensity values in $[0, 255]$. LDOF trajectories do not directly report visibility; the correspondences for trajectories after occlusions are simply missing. These entries are ignored as if they had $\nu_p(t) = 0$. We use the location and frame of the first observation of each trajectory as its reference appearance. LME paths are anchored either in the first or last frame and do report visibility values.

We aim to recover paths that maintain correspondence across occlusions. We measure our success by analyzing the average length of a path, defined as $\frac{1}{N}\sum_p \sum_t \nu_p(t)$. As can be seen in Table 6, the two methods that estimate visibility (our method and LME) return significantly longer paths on average as the result of their ability to detect disocclusions. Further, the average length of our paths tends to correspond to the length of the dominant occluder.

A key feature of our algorithm is the ability to compute the path for every visible point in a scene. We measure path density by computing the distance to the closest visible path for each pixel in the sequence. Table 6 reports the 50th, 95th, and 99th percentile for each method, as well as the total percentage of pixels with a visible path within a distance of 1 pixel. LDOF trajectories leave many pixels unexplained because they are not initialized in low-texture areas. LME misses objects not visible in either the first or last frame of a sequence. In many sequences, these missed objects can account for a significant fraction of the scene. Our method explains over 97% of the pixels in every sequence.

Parameter sensitivity Our technique uses a few parameters that could be tuned if desired. We selected settings for the parameters by hand considering the results on the flowerbed sequence only, and used the same values for all five sequences. We set $\lambda = 1$, $\sigma = 50$, and $\lambda_L = \lambda_T = \lambda_S = 0.5$. We re-scale intensity values to $[0, 1]$ for the combinatorial optimization step to match the range of the binary unknowns. In our experiments, we found that results were relatively insensitive to small changes in the values of λ or σ, but were more sensitive to the values of the parameters for the occlusion detection step.